THE UNIVERSITY OF ALBERTA

PULSE TESTING WITH AN ON-LINE
DIGITAL COMPUTER

BY

ROBERT [illegible]

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
PROCESS CONTROL (CHEMICAL ENGINEERING)

DEPARTMENT OF CHEMICAL AND PETROLEUM ENGINEERING

EDMONTON, ALBERTA
FALL, 1968

THE UNIVERSITY OF ALBERTA

PULSE TESTING WITH AN ON-LINE DIGITAL COMPUTER

BY

ROBERT SCOTT LEES

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

PROCESS CONTROL (CHEMICAL ENGINEERING)

DEPARTMENT OF CHEMICAL AND PETROLEUM ENGINEERING

EDMONTON, ALBERTA

FALL, 1969

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled PULSE TESTING WITH AN ON-LINE DIGITAL COMPUTER submitted by Robert Scott Lees, B.Sc. in partial fulfilment of the requirements for the degree of Master of Science in Process Control (Chemical Engineering).

To

The furtherance of improved
process control in industry

ABSTRACT

Pulse testing is investigated theoretically and experimentally using an IBM 1800 Data Acquisition and Control System. A double pipe countercurrent water - water heat exchanger is used for the experimental investigation.

Considerable material in the thesis is devoted to the description of the hardware and software interface between the IBM 1800 and the heat exchanger system. Direct digital control was used for the operation of the heat exchanger and the computing capabilities of the 1800 enabled a complete pulse test to be directed from a typewriter console within the digital computer laboratory. A complete pulse test involved:

1. control of the process and calculation of the heat balance until steady state operation was ascertained
2. manipulation of a control valve on the process to introduce the desired pulse
3. rapid acquisition of data from process variables required
4. analysis of pulse data to determine the frequency response of the process using three different techniques to evaluate the Fourier transform
5. plotting of the frequency response diagrams of the process

The theoretical investigation demonstrated the effect of the input pulse duration on the frequency response results obtained from a first order system. Also included is an example of two cascaded first order systems and

an underdamped second order system. The effect of an error of closure on the output pulse and the use of spectral smoothing are considered briefly.

Generally, studies were not detailed in any aspect, so few specific conclusions could be drawn. However, the material presented represents a good foundation from which a number of future projects can proceed.

ACKNOWLEDGMENT

The author would first like to express his sincere appreciation to his supervisor, Dr. R. K. Wood, for his guidance and encouragement. The financial support during the project provided from National Research Council Grant A-1944 is gratefully acknowledged. The help of the Department of Chemical and Petroleum Engineering's DACS Centre staff is also gratefully acknowledged for without their assistance this project would not have been possible.

Thanks is due to a number of fellow graduate students for their ideas and suggestions. A particular thank you is due to R. B. Newell for his valuable assistance.

Finally, the author would like to express his deepest appreciation to his wife, Vanna, for her understanding, encouragement and enthusiasm when it was needed most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

In this work a large number of graphs were obtained using a computer plotting routine. These graphs do not have titles and, therefore, are not classified as figures. Throughout the thesis there are tables listing the graphs presented and describing the significance of the information presented in each case.

I. INTRODUCTION

1.1 Introduction

Over the last fifteen to twenty years industrial process control has evolved from an art into the complex science it is today.

In the past, process units were designed to meet steady state performance criteria. Little or no consideration was given to the dynamic aspects which are so vitally important in determining the performance under automatic control. Usually the control instruments used were purchased on a rule of thumb basis or on a suppliers recommendation. They were "tacked" on to the process unit and adjusted in a trial and error fashion until a satisfactory degree of control was obtained. Occasionally, the dynamic design of the process unit was extremely poor and the built in flexibility of the control instruments was insufficient to compensate. In such cases, satisfactory control could not be obtained so the instruments were usually operated in manual mode and considered to be poorly designed.

Today the general knowledge and understanding of automatic control theory is at a much higher level. It is generally realized that the process unit is an important element of any control loop and consequently there is a trend toward increased consideration of dynamics in process design. The instrumentation that is now available is much more sophisticated and now makes possible the implementation of advanced control theory. The most promising example of sophisticated equipment is the process control computer which offers an almost unlimited latitude for implementation of control logic.

However, in spite of the great strides in control theory and control equipment design, there still exists a problem area in the field of process control. It is the serious shortage of dynamic mathematical process models which are generally required for development and design of control loops. The shortage of information in this area can be attributed to the complex interactions of process variables and the individual nature of process units which make models derived from theoretical considerations almost impossible to obtain. Or, if a theoretical model can be obtained, it is often too complex to be used for control purposes. As a consequence of the difficulties associated with the theoretical approach to process modelling, there has been a considerable amount of work directed toward the development of empirical models using the results of dynamic tests conducted in a full scale plant or on a prototype model. A number of methods for this type of testing have come into common usage and among these is the pulse testing techniques studied in this work.

If the current trend toward implementation of computer control continues in the future, it is reasonable to assume that the requirement for dynamic mathematical process models must also increase. Consequently, there is a requirement today for a dynamic testing method which can be used successfully by the practicing process engineer at a minimum cost for test equipment and time. It is the author's opinion that the pulse testing technique studied in this work satisfies this requirement.

1.2 Methods of Dynamic Analysis

The dynamic characteristics of a process unit can be considered known when the time varying functional relationships between the inputs

and the outputs are defined. There are a number of methods in common usage today which are suitable for determination of these relationships. On simple systems it is often possible to write the differential equations which describe the dynamic relationships. However, as mentioned previously, for most industrial process units complexity of the functional relationships makes dynamic testing the only feasible approach to determining dynamic characteristics.

In general, this type of testing involves forcing one of the process unit inputs in a fashion which will excite the unit dynamics. Any other inputs to the unit are held constant and the effect of the excitation is monitored by recording the time varying response of all process unit output variables.

Most dynamic testing methods may be categorized into the following four areas on the basis of the type of input forcing function used:

a. Initial condition inputs

b. Sinusoidal inputs

c. Random inputs

d. Pulse inputs

The initial condition input is usually a step change and the information yielded is usually evaluated in the time domain. This method of analysis is commonly referred to as transient response analysis as opposed to the frequency response analysis which takes place in the frequency domain. Although transient and frequency response analyses are related for linear systems, they are often considered independently. In the four categories above, information from initial condition inputs is usually the only type analyzed in the time domain. The other methods are generally associated

with frequency response, although the pulse and initial condition categories are somewhat interchangeable. For example, a long duration pulse may be used where it is not desirable for the process to be away from the operating point indefinitely as it would be on a step change. Also, it is theoretically possible to extract frequency response information from the transient data obtained with a step input, although practical considerations indicate other types of inputs are more suitable. (59, 70)

In the area of transient analysis from initial condition forcing, it has been common practice for process personnel to use the step input as a means to gain intuitive knowledge of the process dynamics. However, more quantitative information is now being obtained through transient analysis. Shinskey (72), in a recent book, has suggested that significant quantitative information can be learned about the dynamics of a process unit by studying the equipment configuration and conducting a single open and closed loop test with a conventional analog controller. Sanborn (68) reported the successful application of a technique where simple first and second order systems with dead time were used as models for complex industrial processes. Basically, the technique involved evaluation of the model parameters so the transient response of the process agreed with that of the model on the basis of a best fit least squares criterion. This technique has been extended into the design of digital control algorithms using what is called the cancellation compensation method. This material, however, is beyond the scope of this work and will not be discussed.

The use of a sinusoidal input of constant amplitude and frequency is often referred to as the classical frequency response method. The outputs of the process unit when subjected to such an input will appear as an oscillatory signal which will be a pure sine wave if the process is linear. When the transients in the output disappear, the amplitude ratio of the output to the input and the phase difference between the two signals are recorded. These two values constitute one point on the frequency response diagram for the process being tested. Other points on this diagram are determined by repeating the procedure outlined using a different frequency sine wave input for each test. This method of dynamic testing, although commonly understood, is not commonly used to obtain dynamic information about industrial processes. There are three main reasons for its lack of popularity:

a. The field testing time is long and tedious. Likewise, the analysis of the data can also require considerable time, especially if the signals are noisy.

b. A sinusoid on the process input which can be detected at the output is usually sufficiently large in amplitude to cause off specification production. This in conjunction with the length of field testing time leads production requirements to forbid such tests.

c. Pure sine wave actuators are difficult and expensive to build for many process inputs.

The random input analysis almost certainly requires an on-line digital computer before it is feasible to use. It involves reading large numbers of data points from the process unit input and output. From this data statistical auto and cross-correlations and Fourier transforms are calculated to yield frequency response data. The

theoretical advantage of this method is that it does not require large magnitude input disturbances to yield dynamic information. However, Wollaston (78) indicated it is most unlikely that natural statistical inputs will meet the desired criteria, so generated random inputs must be applied. It is reasonable to assume that a random actuator for a process input variable would be no easier, and maybe more difficult, to build than a pure sine wave actuator. It, therefore, follows that the requirement for a generated random input may seriously hamper the popularity of random noise analysis, even if a computer is coupled to the process.

A pulse input is defined as any signal which differs from zero or some reference value for a finite period of time. The strength of a pulse input must be sufficient to excite the process unit dynamics without driving the process unit beyond its capacity to respond. The shape of the pulse may be entirely arbitrary, although several mathematical forms have been investigated to provide general guidelines on the effect of shape. Information derived from a single properly executed pulse test is equivalent to that which can be obtained from a series of sinusoidal inputs of different frequencies. This is because a pulse is composed of an infinite number of harmonic frequencies which excite the process unit simultaneously. Determination of the process unit response to each of these frequencies is readily obtained from the Fourier transforms of the input and output pulses. The Fourier transform calculation can be easily programmed on most general purpose digital computers. Some advantages of pulse testing are the following:

a. Minimum plant testing time, since one properly executed pulse test is sufficient.

b. No requirement for sophisticated pulse generation equipment, e.g. a simple hand valve can be used for a flow pulse.

c. Once the Fourier transform program is obtained, the analysis of the data requires little labour.

d. The technique is suitable for testing equipment which can only tolerate short term deviations of the input variable from the normal operating point. This applies to equipment ranging from volatile chemical plants to air craft in flight.

The literature survey did not reveal a reported application of pulse testing where useful information was not obtained. Also, literature reports based on experience with pulse testing and other techniques generally indicated that pulse testing showed the most promise. It is significant that the Instrument Society of America recently adopted pulse testing as a standard dynamic testing technique. (17)

1.3 Selection of Equipment

To properly evaluate pulse testing, it was necessary to have a physical system which could be used as a "test bench". It was decided that the concentric double pipe heat exchanger would be the best choice for the following reasons:

a. The heat exchanger is a process unit which is common to almost all industrial plants.

b. The published literature on heat exchanger design, steady state and dynamic characteristics and control is extensive. In particular, for the double pipe heat exchanger analytical

non-linear and linearized models have been simulated using digital and analog techniques, and the results of these simulations have been correlated with experimental data. It was felt that this large amount of information would be extremely useful for quantitative evaluation of the dynamic data obtained by pulse testing.

c. The double pipe heat exchanger is relatively simple to build and can be easily started up and shut down over short periods of time.

II. LITERATURE SURVEY

2.1 Classification of Literature

The published literature reported in this chapter has been divided into two broad categories. The first category includes the literature exclusively related to heat exchanger dynamics, while the second includes that related to pulse testing. The studies on heat exchanger dynamics involving pulse testing have been placed in the second category.

The literature on heat exchanger dynamics in the first category is further sub-divided into investigations concerned with temperature forcing and flow forcing. The publications considered in each of these categories are presented in chronological order. The literature on pulse testing in the second category is also presented in chronological order.

2.2 Heat Exchanger Dynamics

2.2.1 Temperature Forced

The earliest investigators into heat exchanger dynamics concerned themselves with the response to temperature forcing. This was because the differential equations describing these responses were linear with constant coefficients provided that the heat transfer coefficients could be considered independent of temperature, which was considered a reasonable assumption.

Williams and Morris (80) reviewed most of the work conducted on heat exchanger dynamics through the 1950's. Rosenbrock (67) published a literature review on heat exchanger and distillation column dynamics in 1962.

Debolt (21) was one of the first people to observe the resonance phenomena on a temperature forced heat exchanger. Cohen and Johnson (16) predicted resonance theoretically using a distributed parameter model. However, their experimental data did not extend to high enough frequencies. In 1961 they published a second study (15) which experimentally verified their prediction of resonance.

Paynter and Takahashi (61) developed exact solutions for the frequency response of models of temperature forced concentric tube heat exchangers. Both parallel and counter flow were considered.

Hsu and Gilbert (44) considered theoretical transfer functions for various types of temperature forced heat exchangers. Their work was entirely theoretical, however, references were made to previous experimental work.

Harriott (34) and Buckley (10) in their textbooks discuss some of the more recent reports on the dynamics of temperature forced heat exchangers.

Mozely (58) was one of the first to use a lumped parameter model for a liquid - liquid heat exchanger. He compared his model to experimental data derived from sinusoidal temperature forcing of a heat exchanger and a passive analog model.

Thal-Larsen (76) neglected the complexities of theoretical models and developed several simple models from knowledge of the basic characteristics of heat exchangers. He verified his model for the concentric pipe heat exchanger by correlating it with frequency response diagrams determined experimentally by other workers. He concluded that a single basic parameter may be used to dynamically characterize entire

families of heat exchangers.

Chan and Leonard (11) have presented theoretical research on the dynamics of temperature forced, parallel flow heat exchangers with no phase change.

2.2.2 Flow Forced

Lees and Hougen (52) and Hearn (36) reported the only studies of flow forced disturbances prior to 1970. Because they used pulse testing the details of their work are contained in Section 2.3.

Edwards (26) was one of the first to study flow forcing of liquid - liquid heat exchangers. He developed his model for countercurrent flow. Resonanace was not reported.

Hempel (38) used a steam - water heat exchanger and derived transfer functions for temperature forcing of steam and water as well as flow forcing of the water. Resonance was predicted theoretically and demonstrated experimentally.

Koppel (49) obtained an analytical solution to a non-linear model consisting of a single partial differential equation. He compared this solution to a linearized version of the same model to determine the effect of linearization.

Isom (46) studied theoretically and experimentally a flow forced liquid - liquid double-pipe heat exchanger using a shell with and without baffles. He used both distributed parameter and lumped parameter models and simulated the latter on the analog computer. Resonance was observed experimentally, but it was not as obvious as in other research.

Stermole (74, 75) also studied a flow forced concentric tube, liquid - liquid heat exchanger. He used distributed and lumped parameter models. Resonance was observed experimentally.

Herron (37) devoted a major portion of his research towards developing a successful digital simulation of a double-pipe heat exchanger. He considered countercurrent flows and a liquid -liquid media. The experimental investigation was very limited.

Fisher (28) experimentally investigated the affect of Reynolds Number, Prandtl Number and heat exchanger size using sinusoidal flow forcing on a liquid - liquid, countercurrent concentric pipe heat exchanger. A frequency response diagram was used to compare the experimental data with values predicted from a mathematical model. Resonance was observed and discussed.

Privott (62) studied a concentric tube liquid - liquid heat exchanger using flow forcing. He simulated a non-linear model, both digitally and with an analog computer. The non-linear model results when compared to data from an experimental heat exchanger were found to be in good agreement. Using perturbation techniques, a linearized model was derived from the non-linear model and the effect of linearization was studied in both the time and frequency domains. Privott is one of the few researchers who used a model which did not neglect the heat capacity of the shell walls.

2.3 Pulse Testing

To the best of the author's knowledge, the first application of pulse testing of physical systems took place in the air craft industry in the early 1950's. Walters and Rea (79), Seamans, Blasingame and

Clementson (71), Rea (64), Boilay (6) and Simth and Triplett (73) all published work related to the pulse testing of air craft while in flight. Draper, McKay and Lees (22) published a book which, among other things, provided one of the first complete reports on the mathematics of pulse testing. The techniques they present for evaluating the Fourier transform have been used extensively and are still popular today.

Lees and Hougen (52) demonstrated the applicability of pulse testing to heat exchangers. Using flow forcing of a steam - water heat exchanger, they were able to obtain good agreement between the experimental and analytical results when compared on a frequency response diagram. This work was extended by Morris (57), Hearn (36) and Vincent (77). Morris (57) used pulse testing to obtain the transfer function relating the tube outlet temperature to changes in the tube inlet temperature on a commercial vapor - liquid heat exchanger. Hearn (36) pulsed the shell flow on a water - water baffled heat exchanger and obtained a linear model from the frequency response diagram by using standard linear profiles presented in Draper, McKay and Lees (22). Vincent, Hougen and Dreifke (77) used pulse testing to study the dynamics of fluid mixing in shell and tube heat exchangers by introducing a temperature pulse in the shell side with the tubes being devoid of fluid. From their results they concluded that mixing is the predominant mechanism determining the effluent fluid temperatures of shell tube heat exchangers.

In 1960, Head, Hougen and Walsh (35) demonstrated the feasibility of pulse testing for procuring dynamic information on complex processes by studying dynamics of the dispersion of solutes in flowing liquid systems. In the same year Grantom, Hougen and Driefke (33) used pulse

testing to study the dynamic lags in stream analyzers.

By the late 1950's pulse testing had become fairly common in the process industries and general reviews of the method appeared in the literature. Lees (51) published a general review of dynamic analysis presenting several methods, including pulse testing. In 1961, Hougen (41) published a general discussion on pulse testing presenting the technique in a simple condensed form. In the same year Hougen and Walsh (42) co-authored a very comprehensive review of the pulse testing methods. They presented nine applications of pulse testing to physical systems, some of which were taken from the papers reported in this survey.

Driefke (25) did an extensive study into the effect of pulse shape using analog simulation of system up to the sixth order with three resonant peaks. In all cases he was able to select a single pulse which yielded a frequency response curve that agreed very well with the theoretical curve. Driefke was also able to demonstrate that valid information could be obtained between the zeros of the frequency content curve of the input pulse.

In 1962, Driefke, Hougen and Mesmer (24) studied the effect of an error of closure on the output pulse using pulse time data derived from an analog simulation of a second order system with various damping ratios. They concluded that some reliable results can be obtained under truncation. However, it can be seen from their results that the data scatter at higher frequencies increases with an increased amount of truncation.

Clements (12, 13) also studied the effects of pulse shape and by means of a graph clearly illustrated the relationship between pulse shape and frequency content. Using data from an analog computer, he also studied the effect of the error of closure in the output pulse, the effect of the number of significant figures in the time data, the effect of using too few data points to approximate time curves and the effect of using a variable integration increment for the time curves. Clements used pulse testing experimentally to determine the dynamics of a packed extraction column.

Hougen (40) published an excellent review of dynamic analysis which includes a comprehensive section on pulse testing. He provided an excellent summary of the effect of pulse shape which is presented in part in Section 3.2.2 of this thesis. Also, he introduced a technique whereby the specification for a closed pulse can be dropped by using developments presented in the work of Schechter and Wissler (70) and Nyquist, Schindler and Gilbert (59). Both of these studies were concerned with the determination of frequency response information from a step response.

Banham (1) applied pulse testing to a full scale naval steam generating system after failing to obtain satisfactory dynamic information using transient and pure frequency forcing. He applied an analog filter to reduce spectral noise and used a specially built analog computer to calculate the Fourier transform. He was able to obtain good agreement between theory and experiment and claimed an economic advantage in using pulse testing over transient or pure frequency forcing.

To the author's knowledge, the most up-to-date general review of pulse testing is that of Driefke and Hougen (23). Their report is

highlighted by a good development of the mathematics of pulse testing. They discuss features which should be part of any computer program used to analyze the data and mention the necessity of having a routine to handle pulses which do not close. Examples of the application of pulse testing to equipment ranging from a pressure transducer, which required an oscilloscope to record the response to a 200-plate distillation column which had an output pulse duration of 138 hours are given.

Lees and Dougherty (50) investigated some of the problems of numerical analysis in determining the Fourier transform.

Janis (47), Fogle (29), Go (32), Renfro (65), and Marino and Stutzman (54) in the period from 1965 to 1967, have applied pulse testing in the study of the dynamics of distillation columns.

Wollaston and Swanson (78) compared the pulse, step and random varying forcing functions. When applied to different chemical processes simulated on an analog computer. They introduced a new lag window to reduce spectral noise. They concluded that if the process noise level is not excessive and if there are no restraints on the process input, the pulse signal is the best to use.

Marina, Perna and Stutzman (53) compared the frequency response results determined from the response of a 24-plate distillation column to sinusoidal and pulse forcing of the reflux rate. They used results obtained by sinusoidal forcing as a standard on which to evaluate the validity of pulse testing. They considered pulse test results valid when they agreed within 25%.

The wide acceptance of pulse testing is borne out by the fact that in October 1968 the Instrument Society of America published a standard on dynamic testing (17) which included a section devoted to pulse testing.

III. MATHEMATICAL BACKGROUND OF PULSE TESTING

3.1 Introduction

In 1822, Fourier introduced the Fourier series (30). This series, which is the basis of the mathematics in pulse testing, provided a means of representing periodic functions in terms of a series of sinusoidal waves of different frequencies and amplitudes. In the case of a single pulse the period is infinite and the Fourier series representation becomes what is known as the Fourier transform. When the Fourier transform of the input and output pulses obtained from a pulse test are evaluated, the frequency response of the system pulse tested can be readily obtained. Until the development of the digital computer the number of calculations involved in evaluating the Fourier transform of transient data was prohibitive for most practical cases. However, today this is no longer a problem and pulse testing has experienced a tremendous growth in popularity.

The information contained in this chapter provides a general description of the manner in which the Fourier transform is used in pulse testing. Some of the material in this chapter has been taken directly from Clements (12).

3.2 Application of the Fourier Transform to Pulse Testing

3.2.1 General

The transfer function of a linear process may be defined as the ratio of the Laplace transform of the process output to the Laplace transform of the input, all initial conditions being zero.

Thus, from the definition of the Laplace transform the transfer function is given by:

$$G(s) = \frac{\mathcal{L}[y(t)]}{\mathcal{L}[x(t)]} = \frac{\int_0^\infty y(t)\ e^{-st}\ dt}{\int_0^\infty x(t)\ e^{-st}\ dt} \tag{3.1}$$

Where G(s) is the process transfer function

y(t) is the process output

x(t) is the process input

t is time

s is the complex variable of the Laplace transform

$\mathcal{L}[\ ]$ indicates the Laplace transform operation

In the remainder of this work only functions which are identically zero for negative t need be considered.

From consideration of the behaviour of Equation (3.1) along the imaginary axis, provided the transforms do converge for the real part of s equal to zero, it follows that the equation can be stated as

$$G(jw) = \frac{\int_0^\infty y(t)\ e^{-jwt}\ dt}{\int_0^\infty x(t)\ e^{-jwt}\ dt} = \left| G(jw) \right| \exp\left(j \text{ arc } \left[G(jw)\right]\right) \tag{3.2}$$

Obviously, with the above-mentioned restrictions regarding x(t) and y(t), G(jw) is the ratio of the Fourier transforms. These restrictions can be easily satisfied by measuring y(t) and x(t) as perturbations from some steady state values which existed at zero time.

In the remainder of this study the following designations will be employed:

$$\left| G(jw) \right| = AR(w)$$

$$\exp\left(j \text{ arc}\left[G(jw)\right]\right) = \exp j\, \phi\, (w)$$

Equation (3.2) can also be written as:

$$AR\,(w) = \left| \frac{\mathcal{F}[y(t)]}{\mathcal{F}[x(t)]} \right| \tag{3.3}$$

$$\phi\,(w) = \text{arc}\ \frac{y(t)}{x(t)} = \text{arc}\ y(t) - \text{arc}\ x(t)\ (2\pi) \tag{3.4}$$

$\mathcal{F}[\ \ ]$ indicates the Fourier transform operation.

AR = amplitude ratio of the process

ϕ = phase lag of the process

In other words, if the integrals in Equation (3.2) can be evaluated numerically from experimentally obtained pulse data for x(t) and y(t), then the process frequency response can be readily calculated from Equation (3.3) and (3.4).

3.2.2 Use of the Frequency Content Curve

The frequency content curve for the input pulse is of paramount importance in assessing the frequency response information obtainable from a pulse test. This curve is a graph of the magnitude of the Fourier transform of the input pulse, $\left| \mathcal{F}\ x(t) \right|$, versus frequency. The ordinate of the frequency content curve at any given frequency is proportional to the magnitude of the sinusoidal content of the pulse at that frequency. At frequencies where the magnitude of the Fourier transform becomes small, in the vicinity of the zeros, valid frequency

response information cannot be determined. Evidence of such difficulty appears as discontinuities in the frequency response diagram, established using Equation (3.2). In mathematical terms, a low or zero frequency content amplitude implies division by a small number or zero in this equation.

The input pulse utilized should provide a sufficiently high frequency content that is, a high Fourier transform magnitude, at the frequencies of interest. In general, the shape of the frequency content curve is a function of the duration and shape of the pulse. For example, a rectangular pulse has a frequency content curve of the form shown in Figure 1.

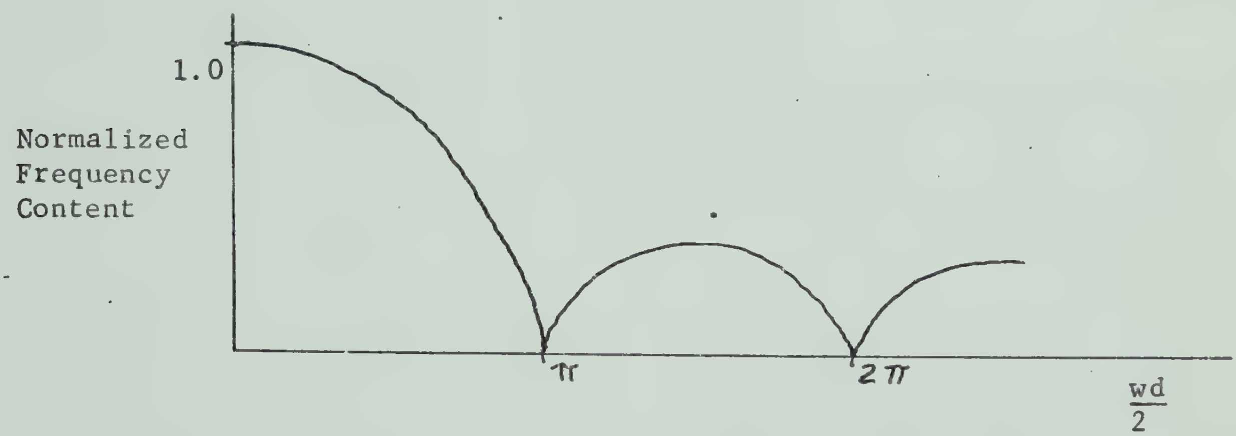


Figure 1 - Frequency Content of a Rectangular Pulse

The following points concerning Figure 1 should be noted:

a. The frequency content is normalized on the basis of the content at zero frequency.

b. Only positive frequencies are considered.

From Figure 1, it can be seen that the first zero in the ordinate of the frequency content curve occurs when $wd/2 = \pi$ or $w = 2\pi/d$. Therefore, the narrower the pulse, (d = duration) the higher the value of the frequency w at which the ordinate of the frequency content first becomes zero. Since the curve decreases monotonically to the first zero, a more desirable frequency content curve will result when this zero occurs at the highest frequency possible. The ideal frequency content curve results from an impulse which has zero width. For this case the frequency content curve is a constant value of 1.0 at all frequencies. In general, all shapes of one-sided pulses exhibit a decay in frequency content ordinate with increasing frequency and the rate of decay increases as pulse duration increases.

The relationship between frequency content and pulse shape has been studied by a number of investigators, notably Clements (12), Dreifke (25) and Lees (51). The pulse shapes which have been investigated may be placed in two categories; one and two-sided. One-sided pulses investigated include primarily the rectangular, triangular, displaced cosine, weighted displaced cosine, cosine, ramp, squared ramp. Two-sided pulses include the rectangular doublet, full sine damped sine and others. Study of the results of Driefke (25) and Lees (51) led Hougen (41) to the following conclusions:

a. One-sided pulses which are symmetrical about their maxima all yield identical Fourier transform phase angles, that is, $(-wT)/(2)$ radians.

b. Pulses such as a rectangular doublet or full sine wave have phase angles which lag those of one-sided pulses by 90 degrees.

c. All one-sided pulses exhibit their maximum frequency content at zero frequency, and those which are symmetrical about the maximum pulse amplitude exhibit one or more zeros in their frequency content.

d. As the one-sided symmetrical pulses become smoother in their initial and final first derivatives, the magnitude of the frequency content is maintained at higher levels over a wider frequency range.

e. Unsymmetrical one-sided pulses may have no zeros in their frequency content.

f. As one-sided pulses become more and more unsymmetrical by returning more quickly to zero than they depart from it, their frequency content amplitude diminishes less rapidly. The ultimate is displayed by the ramp pulse.

g. Two-sided equal-weighted pulses, such as the rectangular and full sine have a frequency content of zero at zero frequency and exhibit a maximum at some higher frequency.

h. A damped one-cycle sine wave pulse has a finite frequency content at zero frequency but exhibits its maximum value at a higher frequency.

The implication to be drawn from Hougen's last two observations is that pulses having properties similar to the last three can be used for better system excitation in selected frequency ranges. These forms should be particularly useful for diagnostic purposes.

Disregarding the effect of shape, it would appear from the discussion on width of one-side pulses, that the narrowest input pulse physically obtainable would be the pulse to use in a pulse test because it would place the first zero of the frequency content at the highest possible frequency. However, pulse width cannot be chosen arbitrarily because a very narrow pulse of finite height cannot generally drive a process sufficiently hard to generate a time response that is large enough to be distinguishable from the process noise. The driving force of a pulse is a function of its area. In many cases, the height is often limited by physical constraints in the process tested. For example, it is meaningless to attempt to drive a process so hard that its capacity to respond is exceeded or until large non-linearities are introduced. Therefore, for a given process the height of pulse is generally fixed and the necessary driving force must be obtained by increasing the pulse duration. However, such action decreases the frequency content by lowering the frequency at which the first zero occurs. In most cases a compromise must be made between the requirements for driving force and frequency content. Selecting a pulse with the "optimum" shape can partially compensate for the loss in frequency content through increasing the duration. However, most often it is the experience of the person conducting the test that finally determines the pulse shape, duration and amplitude that will be used.

In the case of one-sided pulses, which are symmetrical about the maximum pulse amplitude, the frequency content curve is

characterized by one or more zeros. The classic case is the rectangular pulse for which the frequency content curve was shown in Figure 1. As discussed previously, the frequency response diagram exhibits discontinuities at frequencies where the amplitude of the frequency content curve is low or zero. However, the amplitude of the frequency content curve for symmetric pulses rises and falls between the zeros and the amplitude on each "hump" gradually decays as the frequency increases. Theoretically, it is possible to derive valid frequency response information between the zeros where the amplitude of these "humps" is sufficiently large to not affect results.

Driefke (25) used the information derived between the zeros and neglected the erratic data calculated near the zeros. Hougen (41) indicated that Driefke (25) had conclusively demonstrated that good results could be computed on either side or between the frequencies at which the input pulse exhibits a zero in its frequency content. However, Clements (12) stated "Although it is sometimes possible to recover reliable frequency response information between zeros of the frequency content, the author has found this is the exception rather than the rule for most practical tests, and it is not recommended that the magnitude ratio and phase lag values be trusted past the first zero." These conflicting opinions will be considered later in connection with the results of this study.

3.2.3 Derivation of Equations to Evaluate the Fourier Transform

The following treatment is restricted to the case where the process input x(t) begins at zero time and is closed, i.e., it is zero at $t < o$,

assumes a finite value for a finite time interval, T1, and returns to zero at the end of the interval. Further, the assumption is made that the process output response y(t) is of the same form, existing only for $0 \leq t \leq T2$, after which it returns to its steady state value so closely that any difference can no longer be distinguished. With these assumptions Equation (3.2) becomes

$$G(jw) = \frac{\int_0^{T2} y(t)\, e^{-jwt}\, dt}{\int_0^{T1} x(t)\, e^{-jwt}\, dt} \tag{3.5}$$

Using the identity $e^{-jwt} = \cos(wt) - j \sin(wt)$ Equation (3.5) becomes

$$G(jw) = \frac{\int_0^{T2} y(t) \cos(wt)\, dt - j \int_0^{T2} y(t) \sin(wt)\, dt}{\int_0^{T1} x(t) \cos(wt)\, dt - j \int_0^{T1} x(t) \sin(wt)\, dt} \tag{3.6}$$

where the integration limits need extend only over the actual pulse widths T1 and T2. For convenience, Equation (3.6) can be written as:

$$G(jw) = \frac{A - j\,B}{C - j\,D} \tag{3.7}$$

where

$$A = \int_0^{T2} y(t) \cos(wt)\, dt \tag{3.8}$$

$$B = \int_0^{T2} y(t) \sin(wt)\, dt \tag{3.9}$$

$$C = \int_{o}^{T1} x(t)\ \cos(wt)\ dt \tag{3.10}$$

$$D = \int_{o}^{T1} x(t)\ \sin(wt)\ dt \tag{3.11}$$

To obtain the frequency response diagram for a process which has been pulse tested, it is necessary to obtain the amplitude ratio and phase lag of G(jw). Multiplication of Equation (3.7) by the complex conjugate gives

$$G(jw) = \frac{A - jB}{C - jD} \cdot \frac{C + jD}{D + jD} = \frac{AC + BD}{C^2 + D^2} + j\,\frac{AD - BC}{C^2 + D^2} \tag{3.12}$$

So

$$Re(w) = \frac{AC + BD}{C^2 + D^2} \tag{3.13}$$

$$Im(w) = \frac{AD - BC}{C^2 + D^2} \tag{3.14}$$

where Re(w) and Im(w) are the real and imaginary components of the process transfer function. It then can be shown from the theory of complex variables that the amplitude ratio and the phase lag of the process are given by

$$AR\ (w) = \sqrt{Re^2(w) + Im^2(w)} \tag{3.15}$$

$$\emptyset\ (w) = \tan^{-1} \frac{Im\ (w)}{Re\ (w)} + k\pi \qquad k = 0, 1, 2.... \tag{3.16}$$

The frequency content A(w), of the input pulse is given by $\left| \mathcal{F}[x(t)] \right|$ where $\mathcal{F}[x(t)]$ is the Fourier transform of the input pulse.

$$A(w) = \left| \int_{o}^{T1} x(t)\, e^{-jwt}\, dt \right| \tag{3.17}$$

$$= \left| \int_{o}^{T1} x(t) \cos(wt)\, dt - j \int_{o}^{T1} x(t) \sin(wt)\, dt \right| \tag{e.18}$$

$$= \left| C - jD \right| = \sqrt{C^2 + D^2} \tag{3.19}$$

The normalized frequency content is defined as $A(w)_N = A(w)/A(o)$ so evaluation of Equation (3.18) for w=o yields

$$A(o) = \int_{o}^{T1} x(t)\, dt \tag{3.20}$$

Combination of this expression with Equation (3.19) gives

$$A(w)_N = \frac{\sqrt{C^2 + D^2}}{\int_{o}^{T1} x(t)\, dt} \tag{3.21}$$

The computational problem becomes the numerical evaluation of the integrals A, B, C and D using selected values of w and experimental pulse data. The amplitude ratio (AR) and phase lag (Ø) of the process pulse tested can then be readily determined using Equations (3.15) and (3.16).

3.2.4 Quadrature Formulas

There are a number of methods described in the literature for obtaining numerical approximations to the transcendental integrals A, B, C and D. Draper, McKay and Lees (22) developed a method called the stepped curve or rectangular approximation which offers the advantage of simplicity. Also, they presented a form of the trapezoidal approximation which was simplified by neglecting the first and last ordinates on the assumption they would be zero, i.e. the pulse was assumed to be closed. This modified trapezoidal approximation is considered to be a refinement over the stepped curve approximation and it has been used in a large number of the investigations reported in the literature.

Dreifke (25) developed a trapezoidal approximation called TAFT which he felt offered a slight advantage over the methods suggested by Draper, McKay and Lees (22).

Clements (12) used a trapezoidal approximation similar to the one introduced by Draper, McKay and Lees and a special quadrature developed by Filon (27). The problem of using quadratures to approximate the transcendental integrals A, B, C and D at higher frequencies was also discussed. Generally speaking, the product curves (x(t)sin (wt), x(t)cos(wt), etc.) oscillate so fast for large values of w that most quadrature formulas break down unless minute integration steps are used. Clements described some correction functions which have been used in the past to compensate for inaccuracies which result when an inadequate number of samples are used to approximate these rapidly

oscillating curves. However, Lees and Dougherty (50) in 1967 concluded that these functions are of little significant value.

In this work three methods for approximating the Fourier transform were used:

a. Trapezoidal

b. Filon

c. Fast Fourier

The application of each of these methods is discussed in the same order in the following text.

The trapezoidal approximation used is the same as that used by Clements (12). It simply involves applying the trapezoidal rule

$$\int_a^b f(t)\,dt \approx \Delta t\,(f_o/2 + f_1 + f_2 \text{ ------------ } + f_{n-1} + f_n/2)$$

directly to the integrals in Equations (3.8) to (3.11) inclusive. Where Δt is the sample interval used for f(t).

f_i, i = 1, 2 ------ n, are the ordinate values of f(t) at each sample interval

If the pulse is assumed closed, the values of fo and fn can be considered zero and Equations (3.8) to (3.11) can be written in the following form for easy evaluation using a computer.

$$A = \Delta t2 \sum_{k=1}^{n-1} y\,(k\,\Delta t2)\,\cos(wk\,\Delta t2) \qquad (3.22)$$

$$B = \Delta t2 \sum_{k=1}^{n-1} y\,(k\Delta t2)\,\sin(wk\Delta t2) \qquad (3.23)$$

$$C = \Delta t1 \sum_{i=1}^{m-1} x\,(i\Delta t1)\,\cos(wi\Delta t1) \qquad (3.24)$$

$$D = \Delta t1 \sum_{i=1}^{m-1} x\,(i\Delta t1)\,\sin(wi\Delta t1) \qquad (3.25)$$

Where n = total number of ordinate values used to approximate the output pulse function.

m = total number of ordinate values used to approximate the input pulse function.

The quadrature developed by Filon (27) was designed especially for approximating integrals of the general form $\int_a^b f(t)\sin(wt)\,dt$ and $\int_a^b f(t)\cos(wt)\,dt$. It is based upon an approximation of the pulse curve using parabolic segments as are used in the Simpson's rule approximation. However, the coefficients of Simpson's rule (1/3, 2/3 and 4/3) are replaced by trigonometric functions of $w\Delta t$ where

w = frequency in radians/unit time

Δt = the sample interval used for $f(t)$

The advantage of Filon's quadrature over other more common methods lies in its independence of the value of w. Because the trigonometric integrations are performed analytically, the problem of rapidly oscillating product curves does not exist (12).

When Filon's quadrature is used to evaluate

$\int_a^b x(t)\sin(wt)\,dt$, the area under the curve $x(t)$ is divided into an odd number of subareas at intervals of Δt. These points are denoted as $I_1, I_2,$ ---------- I_{2n+1} where

$$I_1 = x(a)\sin(wa)$$

$$\vdots$$

$$I_{2n+1} = x(b)\sin(wb)$$

and where

$$S_{2n+1} = \tfrac{1}{2}I_1 + I_3 + I_5 \text{ ---------- } + \tfrac{1}{2}I_{2n+1} \tag{3.26}$$

$$S_{2n} = I_2 + I_4 + I_6 + \text{ -------- } I_{2n} \tag{3.27}$$

$$\theta = w\,\Delta t \tag{3.28}$$

$$\alpha = \frac{1}{\theta} + \frac{\sin 2\theta}{2\theta^2} - \frac{2\sin^2\theta}{\theta^3} \tag{3.29}$$

$$\beta = 2\left[\frac{\cos^2\theta + 1}{\theta^2} - \frac{\sin 2\theta}{}\right] \tag{3.30}$$

$$\gamma = 4\left[\frac{\sin\theta}{\theta^3} - \frac{\cos\theta}{\theta^2}\right] \tag{3.31}$$

then

$$\int_a^b x(t)\sin(wt)\,dt = \Delta t\left[\alpha\left[x(a)\cos(wa) - x(b)\cos(wb)\right] + \beta\left[S_{2n+1}\right] + \gamma\left[S_{2n}\right]\right] \tag{3.32}$$

On the other hand, if

$$I_1 = x(a)\cos(wa)$$

$$\vdots$$

$$I_{2n+1} = x(b)\cos(wb)$$

Then

$$\int_a^b x(t)\cos(wt)\,dt = \Delta t\left[\alpha\left[x(a)\sin(wa) - x(b)\sin(wb)\right] + \beta\left[S_{2n+1}\right] + \gamma\left[S_{2n}\right]\right] \tag{3.33}$$

When θ is less than about 0.35 radians, loss of significant digits causes inaccurate computation of α, β and γ, so these quantities have to be determined by a Taylor's series expansion of Equations (3.29) through (3.31). The expansions given by Filon (28) are

$$\alpha = \frac{2\theta^3}{45} - \frac{2\theta^5}{315} + \frac{2\theta^7}{4725} - \ldots. + \ldots. \tag{3.34}$$

$$\beta = \frac{2}{3} + \frac{2\theta^2}{15} - \frac{4\theta^4}{105} + \frac{2\theta^6}{567} - \frac{4\theta^8}{22275} + \ldots. \tag{3.35}$$

$$\gamma = \frac{4}{3} - \frac{2\theta^2}{15} + \frac{\theta^4}{270} - \frac{\theta^6}{11340} + \frac{\theta^8}{997920} - \ldots.. \tag{3.36}$$

When w tends to zero θ tends to zero, so α, β and γ tend to 0, 2/3 and 4/3, respectively, as can be seen from the equations above. In this case, the right hand side of Equation (3.32) reduces to

$$\frac{1}{3} \Delta t \left[2S_{2n+1} + 4S_{2n} \right] \tag{3.37}$$

which is Simpson's approximation.

The above equations related to Filon's approximation can be readily programmed on a digital computer and used to solve the trigonometric integrals A, B, C and D.

The Fast Fourier Transform (FFT) was applied in this work using a computer program supplied by Trinity University Computer Centre (for details see Section 5.4.2.3). The author of this thesis did not investigate the mathematical detail of the FFT, but information is available to the reader in References (9, 14, 18, 19, 31) and (48). The FFT was developed by Cooley and Tukey (19), especially for computing the discrete Fourier transform from a discrete series of time data. The principal advantage of the FFT over more conventional methods lies in a reduced calculation time. Where the trapezoidal and Filon's quadrature require N^2 operations to calculate N frequency points from N time points, the FFT only requires $N \log_2 N$ operations (19). The time saving, however, does not come without restrictions. The value of N for the FFT must be a power of 2 and transform values are only calculated at frequencies given by

$$w = \frac{2\pi i}{N \Delta t} \qquad i = 1, 2, 3, \text{-------} N \tag{3.38}$$

where

N = the number of points processed

Δt = the sampling interval for the time series

The FFT calculates one frequency point for every time point used. However, only half of these values are significant because the higher values lie beyond the folding (or Nyquist) frequency and are aliased by lower frequencies. From Equation (3.38) it can be seen that a large number of points is necessary to obtain a wide range of frequency values and the magnitude of Δt determines the highest frequency value calculated. As Δt becomes larger, with a fixed value for N, the span of frequency values shifts toward the lower end of the frequency spectrum. Consequently, the values of N and Δt must be carefully chosen to ensure that the Fourier transform will be evaluated over the frequency range of interest.

In this work the FFT was used in the same manner as the trapezoidal and Filon quadrature. That is, the real and imaginary parts of the Fourier transform were evaluated for the output and input pulses (integrals A, B, C and D, respectively) and the results were combined into an amplitude ratio and phase lag using Equations (3.13) through (3.16).

3.3 Error of Closure on Output Pulse

Frequently, industrial processes respond slowly to disturbances and, thus, the response to a pulse input exhibits a slow decay to the original steady state. In such cases the duration of the output pulse becomes excessive and the temptation exists to terminate recording the output prematurely creating an error of closure. Since the development of Equation (3.5) assumes that the process has been allowed to return to its original steady state, use of data with an error of closure introduces

inaccuracies in the frequency response results obtained.

Hougen and Walsh (42) report that inaccuracies appear with rather small errors of closure. The significant effect is an advancing of the low frequency results which tends to give steeper frequency response plots and a reduced steady state gain or zero frequency amplitude. Clements (12) made the same observations. Driefke, Hougen and Mesmer (24) reported on the effect of error of closure using data from an analog simulation of a second order system with various damping ratios. Their results demonstrated that as the error of closure increased, the amount of data scatter at the higher frequencies increased.

For some physical processes, a hysteresis or dead band makes closing of the output pulse impossible to achieve. In such a case Dreifke and Hougen (23) recommend the use of a technique which permits elimination of the closed output pulse requirement through some additional data processing. Basically, this technique replaces the closed pulse requirement with a requirement for a zero first derivative on the tail of the pulse. Hougen (40) describes this technique briefly and reports that Nyquist, Schindler and Gilbert (59) have verified it in their extension of Schechter's and Wissler's (70) method of procuring frequency response information from step forcing

3.4 Spectral Smoothing

In noise analysis or spectral analysis, it is common practice to apply filters known as lag windows. These filters act as windows of variable transmission modifying the raw data differently for different lags. The purpose of these filters is generally to reduce the noise in the Fourier transform of a particular time function.

When time is considered a continuous variable and computation is by analog devices, there is an advantage in applying the lag window to the time data before transformation (4). However, when time is considered to be discrete and computation is performed by digital means, it becomes advantageous to transform the data and then apply the smoothing (4). This is because the transforms of the lag windows are sums of Dirac delta functions at appropriate spacings so smoothing can be done by means of convolution (4). This procedure is simple to perform since convolution means smoothing only by weighting factors (4). Hutchinson and Shelton (45) used spectral smoothing applied after transformation because the Fourier transforms were blurred due to truncation of time data. Wollaston and Swanson (78) applied a special spectral window to results obtained by applying pulse, step and random noise inputs to analog simulations of various chemical processes.

In this study spectral smoothing was investigated using the methods outlined by Blackman (4) which were based on the earlier work of Blackman and Tukey (5). Three methods were programmed:

a. Hanning - using the weighting factors 0.25, 0.50, 0.25

b. Hamming - using the weighting factors 0.23, 0.54, 0.23

c. Combination - involves applying the hanning method and then smoothing once more with weights of 0.1t, 0.68, 0.16

The programs used to apply these different methods of smoothing are discussed in Section 5.4.2.4.

IV. THEORETICAL TESTS

4.1 Introduction

In order to study pulse testing and verify the computer programs used to analyze pulse data, it was necessary to obtain some accurate pulse data from known systems. The procedures discussed in this chapter were used to create the data.

4.2 Generation of Theoretical Data

In the past, most, if not all, studies of pulse testing have used off-line computers to analyze the data. With this procedure the pulse data was usually taken from analog recordings and punched on cards for computer processing. In most cases, the test data from known systems was obtained by pulse testing analog simulations of known systems.

In this work the on-line digital computer provided the facility to read pulse data directly from the process and store it in disk files. In order to test the data analysis programs data from known systems was read into the same disk files as were used for storing the experimental pulse test data. The program TSTFR, which is listed in Appendix III, was written for this purpose. As written, the program is limited to a rectangular input pulse which can be used to force a first order, two cascaded first order systems or a second order system. Basically, the program generates the data by evaluating time dependent equations for the input and output pulses at increasing increment times. The equations for the output pulse are those presented by Buckley (10). The program provides complete flexibility with respect to time increment, time

constants, steady state gain, damping ratio, input pulse duration and amplitude.

4.3 First Order System

The procedure used to generate pulse data for a first order system can be determined from the listing of program TSTFR. In this work a first order system with a time constant of 5.0 seconds was used to study the effect of input pulse duration in pulse testing. Using a constant time sample interval of 0.1 seconds, pulse data was generated for input pulse durations of 1, 2.5, 5, 10 and 25 seconds. In addition, data was generated for a pulse duration of 75 seconds with a sample interval of 0.2 seconds.

4.4 Two Cascaded First Order Systems

The procedure used to generate pulse data for two cascaded first order systems can also be determined from the listing of program TSTFR. In this work only one example of pulse testing of cascaded first order systems was included and in this case time constants of 2 and 10 seconds were used.

4.5 Second Order System

The procedure used to generate pulse data for a second order system can also be determined from the listing of program TSTFR. In this work only one example of pulse testing of a second order system was included and it was performed for a damping ratio of 0.1 and a natural frequency of 0.5 radions per second.

It should be noted that the time constant for the second order system (referred to in the listing of program TSTFR) is the reciprocal of the natural frequency of the system.

V. EXPERIMENTAL WORK

5.1 Introduction

The principal objective of this thesis was to investigate the potential of pulse testing as a technique to characterize the dynamics of an industrial process. The process chosen for the investigation was a fluid - fluid concentric tube heat exchanger fabricated from standard copper tubing. It was operated in the countercurrent mode using water for both streams with the shell stream the hotter of the two. Because of time limitations, the dynamic characterization was limited to the response of the shell outlet temperature for changes in the tube flow. However, the heat exchanger was constructed to permit investigation of the other three possible flow-temperature transfer functions.

The operation and testing of the experimental heat exchanger was done entirely using the department's IBM 1800 Data Acquisition and Control System (DACS). Three variables: (a) tube flow, (b) shell flow and (c) shell inlet temperature were controlled by means of the Direct Digital Control (DDC) algorithms available on the IBM 1800. The tube inlet temperature was uncontrolled since the tube fluid was provided by the building domestic water supply which remained reasonably constant.

The computer was programmed so that prior to an experimental run a heat balance for the heat exchanger was calculated at specified intervals of real time. This data was used as a simple but effective means of determining when steady state conditions had been reached.

The experimental pulse was introduced into the tube flow through a control valve manipulated under program control using the computer. The tube flow and the shell outlet temperature response resulting from the valve position change were measured and stored by the computer system at a rate of forty samples per second.

5.2 Equipment

5.2.1 General

The test heat exchanger system was designed to utilize the buildings constant head demineralized water recirculating system. However, a one year delay in the completion of the constant head system forced the use of domestic water and the dumping of the heat exchanger effluent to drain.

Basically, the system, as used, consisted of two separate domestic water supply lines which fed water to the tube and shell of the test heat exchanger which discharged both effluent streams into a common drain. The water for the tube stream was pumped through a control valve and a flow turbine meter before it entered the test heat exchanger. The water for the shell stream was pumped through a flow turbine meter, a steam-water heat exchanger and a control valve before it reached the test heat exchanger. The steam-water heat exchanger was used to provide a controlled hot water temperature at the shell inlet on the test heat exchanger.

A photograph of the test equipment may be seen in Figure 2.

Figure 2 - Test Equipment

5.2.2 Description of Equipment

Figure 3 is a schematic diagram of the test heat exchanger system which lists each component of the system by means of a symbolic name. Where duplication of system components on the tube and shell streams occurs, the symbols end with a 1 or 2 to indicate association with the tube or shell stream respectively. Table 1 contains an alphabetical listing of the symbols shown in Figure 3. For each item there is an explanation of the function of the component, its specifications and its manufacturer. Where information concerning components is desired beyond the scope of that provided in Table 1, the reader will find a list of the manufacturer's addresses in Table IV-A. The following material outlines some of the factors considered in the design and/or selection of the test heat exchanger and associated equipment.

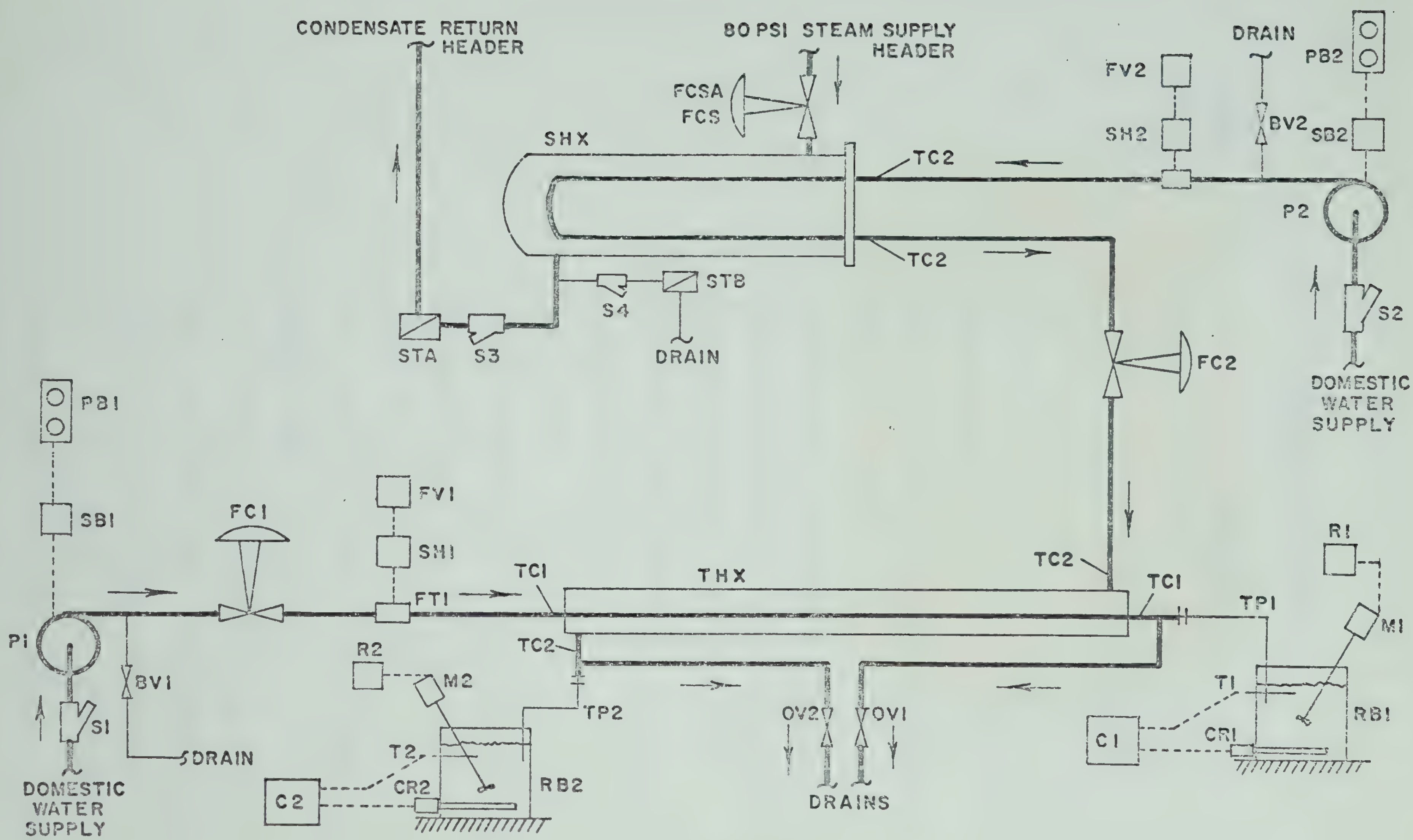


FIGURE 3 - SCHEMATIC DIAGRAM HEAT EXCHANGER SYSTEM
(EQUIPMENT LIST)

TABLE 1

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
BV1	By-pass valve pump 1. (Whitey No. 7RF8, ½" female NPT, brass Cv = 1.1)
BV2	By-pass valve pump 2. (Same as BV1)
C1	On-off temperature controller for reference both on tube outlet. (Assembled at Technical Services Electronics Division as per General Electric Application Note 200.45, April, 1966) (W/O 5/25)
C2	On-off temperature controller for reference bath on shell outlet. (Same as C1)
CR1	Calrod heater for tube outlet reference bath. (Canadian General Electric, 750 watts, 8 inches long, 1 inch male NPT.)
CR2	Calrod heater for tube outlet reference bath. (Same as CR1.)
FC1	Flow control valve for tube stream (3/4" Fisher type 546 - 657A - 3560 control valve, single port, linear valve plug, size 30 actuator, type 3560 valve positioner (3-15 psi) type 546 electro-pneumatic transducer, 4-20 ma) (Serial No. CN 96445A)

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
FC2	Flow control valve for shell stream. (Same as FC 1) (Serial No. CN 96444A)
FC5	Flow control valve for steam to steam heat exchanger. (1½" Fisher type 657A control valve, single port, V-pup valve plug, size 34 actuator 3-15 psi) (Serial No. CN 96447) 10°F superheat Temp. = 348°F P1 = 100 psig, P2 = 80 psig Qmax. = 2400 lb./hr.
FCSA	Positioner for steam flow control valve. (Honeywell model HE 5131 - 1 (685435 -002) electro-pneumatic positioner 4-20 ma.)
FT1	Flow transmitter for tube stream. (Flow Technology, Model FT-12M22-LJ, size 3/4 inch, with 3/4 inch MS-33656-12, 37° male flared tube fittings, 303 s.stl., 430-F s.stl. rotor, and Graphitar #3 journal bearing. Range: 1.5 to 22.0 gpm (US) water at 60°F.)

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
FT2	Flow transmitter for tube stream. (Flow Technology, Model FT-16M60-LJ, size 1 inch with 1 inch MS-33656-16, 37^{o} male flared tube fittings, and material as for FT2. Range: 1.5 to 40.0 gpm (US) water at 60^{o}F.)
FV1 & FV2	Two Channel Pulse Integrator. (Technical Services Department, Electronics Division, University of Alberta. Special device fabricated to integrate pulses from flow turbines and produce a voltage (0 - 5V) proportional to the number of pulses counted) (W/O 5769) (A similar device can be purchased from Flow Technology.)
M1	Mixer motor for tube stream constant temperature bath. (Redmond electric motor model AK4L36J, type AK 4, 1/20 hp, 1.6 amps)
M2	Mixer motor for shell stream constant temperature bath. (Same as M1.)
OV1	Hand valve on outlet of tube stream. (3/4 inch Ladish Globe Valve, Model 7272, with V-Port type plug, travel indicator and flange connections.)

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
OV2	Hand valve on outlet of shell stream (1 inch valve same as OV1)
P1	Pump for tube fluid (Weinman Model 5ACK20P Centrifugal Unipump, with 5 - 1/8 inch diameter impeller 4731, 2 hp 3/60/440 volt 3600 rpm motor, and capacity to deliver 25 USGPM at 105 ft. T.D.H.) (Serial No. 56)
P2	Pump for shell fluid (Weinman Model 6ACK50P Centrifugal Unipump, with 5 - 3/4 inch diameter impeller 4828, 5 HP 3/60/440 volt, 3600 RPM, and a capacity to deliver 12 USGPM water at 135 ft. T.D.H. and 40 USGPM at 75 ft. T.D.H.) (Serial No. 8039 AH)
PB1	Push button to start pump motor on tube stream. (General Electric type CR 104 push button with run indicator light.)
PB2	Push button to start pump motor on shell stream. (Same as PB1)

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
R1	Speed control for mixer motor on reference both on tube stream outlet. (Powerstat type 10B variable autotransformer, $2\frac{1}{4}$ amps.)
R2	Same as R1 but for shell stream. (Same as R1.)
S1	Strainer on water supply of tube stream. (Keckley 2 inch, 125 lb. Y-pattern dirt strainer with 30 mesh screen.)
S2	Strainer on water supply of shell stream. ($1\frac{1}{2}$ inch same as S1.)
S3	Strainer on condensate return line. (Same as S1.)
S4	Strainer on condensate by-pass. (Same as S1 except 3/4 inch.)
SB1	Motor starting box for tube fluid pump. (General Electric CR 106 magnetic motor starter, size DOAAP3.)
SB2	Motor starting box for shell fluid pump. (Same as SB1, except size D1AAP3.)

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
SH1	Shaper amplifier for frequency signal from tube flow turbine. (Flow Technology Inc., Pulse Rate Amplifier, Model PRA - 101, output; square wave 0 to 10 volts, frequency proportional to turbine rpm)
SH2	Shaper amplifier for frequency signal from shell flow turbine. (Same as SH1)
SHX	Steam heat exchanger to generate hot water supply for shell stream. (Taco converter, Model No. 82105, 2 pass, with capacity of 40 gpm (US) with a temperature rise 60^{o}F to 200^{o}F using 80 psi steam.) Inside heat transfer coefficient = 733 BTU/hr.-ft.^{2} - ^{o}F. Outside heat transfer coefficient = 1500 BTU/hr.-ft.^{2}-^{o}F. Overall heat transfer coefficient 492.4 BTU/hr.-ft.^{2}-^{o}F. Heating surface area 38.9 sq. ft. Unit diameter 8" Tubing 3/4" - 18 ga copper Overall length - 6 ft.

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
STA	Steam trap on condensate return line. (Sarco, Type FTS-4C) Float and thermostatic steam trap with 2" ports, capacity of 4720 lbs. of condensate per hour at 75 psi, and a working pressure of 0-125 psi.)
STB	Steam trap on condensate by-pass line. (Sarco, Type FT-125, float and thermostatic steam trap, with 3/4 inch ports, a capacity of 970 pounds of condensate per hour at 75 psi, and a working pressure of 0-125 psi.)
T1	Resistance bulb thermometer. (Selected to be compatible with temperature controller C1 as per General Electric Application Note 200.45, April 1966.)
T2	Resistance Bulb thermometer. (Same as T1 but used with C2.)
TC1	Thermocouples located at strategic points along tube stream. (Thermo-Electric miniature ceramocouple assembles type 5J0110L, with type J calibration, MLP

TABLE 1 (Cont'd)

LIST OF EXPERIMENTAL EQUIPMENT

SYMBOL	DESCRIPTION
	connector and readjustable bushing for 1/16 inch O.D. sheath. Response time for 63.2% of change from room temperature to boiling water 0.09 seconds.)
TC2	Thermocouples located at strategic points along shell stream. (Same as for T1.)
THX	Concentric double pipe heat exchanger. (See text and equipment drawings in Appendix IV for further details.)
TP1	Thermopile on tube outlet. (Scientific Products Corporation, Flexible Cable Thermopile, Model OTP (fast response), with 10 elements in series coated with a 2-inch layer of plastic, overall length 2 feet.) Note: This thermopile failed before tests were completed and a copy had to be fabricated and used. See the text for details.
TP2	Thermopile on shell outlet. (Same as for TP1 but with 25 elements in series) Note: See the note under TP1.

The general size of the test heat exchanger, THX, was determined after a survey of the publications on the dynamic testing of concentric tube heat exchangers. The work which most strongly influenced the choice of size was that of Fisher (28). He showed that as the heat exchanger size (equivalent diameter) increases, the amount of axial mixing decreases so that the experimentally derived amplitude ratio and phase lag curves agree more closely with those determined from a mathematical model which is based upon the assumption of zero axial mixing. To date the majority of the models that have been developed utilize this assumption. In his work, Fisher used the same size of heat exchanger as used by Stermole (74, 75) (1 in. x 2 in. x 14.9 ft.) and obtained the amplitude ratio and phase lag curves using sinusoidal flow forcing. A comparison of experimental results with values predicted from the mathematical model showed close agreement. The size of the heat exchanger used in this work was selected to be of the same order of size as that used by Stermole (74, 75) so that comparison of experimental results with predictions from mathematical models would be possible.

Originally three heat exchangers were built, however, due to lack of time testing was limited to only the largest exchanger. The three heat exchangers were designed to be easily interchanged. Further details on the test heat exchangers are given in Appendix IV.

The Y-strainers (S1 to S4) were installed on the water inlets and condensate outlets as a precaution against debris contaminating critical components of the equipment.

The pumps (P1 and P2) were necessary on both fluid streams to boost the pressure so that high Reynolds numbers could be achieved. The particular pumps used were chosen because of their relatively flat performance curves and they were normally operated on the flattest portion of these curves. In this way, the pumps provided a relatively constant head during the flow disturbance introduced by a pulse test.

The flow control valves (FC1 and FC2), which have linear characteristic plugs, were selected on the basis of the frequency response curves provided by the manufacturer. The amplitude ratio and phase lag were given for sinusoidal forcing amplitudes of 1.5, 5, 10, 20 and 40 percent of span with a mean level of 50 percent of span. The valves are relatively fast acting and were considered to be the best choice for implementing flow pulses of specified shapes through computer generated signals.

Flow turbine meters FT1 and FT2 were used to measure the flow in the tube and shell streams respectively. The signal from the meters was normally used for flow control but during experimental tests the signal was also used for experimental data acquisition. The meter type and manufacturer were chosen on the basis of a recommendation made by Dr. Joel O. Hougen in personal correspondence with the author. Dr. Hougen stated "......... We have used turbine meters extensively for flow measurement. They are especially good for measuring pulses in flow. I would use nothing else ......" He also indicated that the turbine meters chosen for this work had proven quite satisfactory for pulse testing. Specifications supplied by the manufacturer for the turbine meters state that the time constant on

the basis of a step change is three milliseconds.

The turbine meters provide a pulse-like signal with a frequency that is linearly proportional to the flow rate measured. To render this signal more suitable for measurement, shaper amplifiers SH1 and SH2 were also purchased from the turbine manufacturer. These units shape the turbine pulse into a square wave signal of 0 to 10 volts with the same frequency as the original signal. Original plans were to have the computer system count these pulses using software to convert them to the equivalent flow, however, further investigation revealed that the existing DDC package did not provide the facility to control from a pulse count. For this reason, a two channel analog pulse integrator (FV1 - FV2) was built by the Technical Services Department of the University of Alberta to convert the pulse frequency to a proportional voltage between 0 and 5 volts. The slight non-linearity of this unit made it somewhat less than satisfactory for DDC control because a linear conversion from voltage to engineering units was necessary. The resultant errors introduced in the flow rates measured for control purposes were in the order of one percent. This error affected the heat balance calculations since they were performed on the measurements used by the DDC loops. However, the flow data used in the pulse analysis was not affected by the non-linearity because conversion from voltage to flow units was achieved using a third order least squares polynomial fit to the pulse integrator calibration data. (See Section 5.4.2.7 for information on the least squares program.) The calibration data was obtained using a square

wave oscillator, digital frequency meter and digital volt meter. A separate calibration was made for each channel.

The thermopiles TP1 and TP2 were required to detect the small temperature changes resulting from flow pulses. Reports in the literature indicate that construction of these units is rather difficult. Debolt (21) devoted sufficient time in construction of his thermopile to warrant including in his thesis an excellent description of the procedure he used. Because of the possible difficulties in construction, it was decided that the thermopiles used should be purchased from a commercial manufacturer. Two chromel - constantan thermopiles were purchased; one with 25 probes and one with 10 probes. Both were supplied with an overall cable length of two feet with a $\frac{1}{4}$ x 4 inch stainless steel sheath on each end and a flexible plastic sheath in the middle. The two-foot length was chosen because electrical noise increases with the cable length. The sheath at the cold junction end (reference) was closed off so that the probes were not visible. On the hot junction (fast response) end the probes were exposed 3/32" and coated in a plastic material. A seal was obtained at this end by using an expoxy material packed around the probe wires inside the sheath. A single chromel - constanton reference thermocouple was included in the reference end for measurement of the absolute temperature at that point. Four terminal screws on a block were used to connect lead wires for the reference thermocouple and the thermopile.

The thermopiles were mounted with the hot junction immersed in the heat exchanger and the cold junction in the reference bath. A seal was obtained on the heat exchanger stream by means of

a compression fitting supplied with the thermopile. This fitting allowed the depth to which the thermopile probes were inserted into the stream to be adjusted. Since the steel sheath diameters for both thermopiles were identical, it was a simple matter to change the thermopile from one stream to the other.

To secure the thermopile compression fitting in the end of the heat exchanger pipe, two special supports were fabricated. They were designed to perform a secondary function; that of reducing thermal noise which results from hot and cold "pockets" of fluid alternately striking the thermopile probes. Several reports of difficulties with thermal noise from thermopiles appear in the literature. Debolt (21), Cohen and Johnson (15) and Fisher (28) were among those who attempted to solve the thermal noise problem. Fisher's approach seemed the most successful and was successfully applied here. He placed a wad of pot cleaning material upstream to thoroughly mix the fluid before it reached the probes. Details on the supports are given in Appendix IV.

Unfortunately, during the check-out of the heat exchanger system and its hook up with the computer system, the epoxy material surrounding the hot junction probes of the thermopiles began to crack causing a short circuit in the thermopiles. Since the repair period provided by the manufacturer was excessive, a five probe copper constantan thermopile was fabricated as a copy of the manufactured unit. This thermopile was used in conjunction with a chromel - constantan reference thermocouple. The change resulted in programming changes indicated in Section 5.4.2.1 and in program MVTMP shown in Appendix III, but it did not require any changes on the heat exchanger system.

The reference baths (RB1 and RB2) used for the thermopiles were fabricated from an 8-inch length of 8-inch schedule 80 black pipe with the top and bottom cut from ¼" steel plate. The heavy gauge metal added to the heat capacity of the bath so that the on-off controllers, C1 and C2, could easily maintain a constant temperature in the range from 70 to 200°F without insulation. Ethylene glycol was used as the fluid in the baths to eliminate rusting.

Ceramic packed iron constantan thermocouples mounted in a 1/16 inch OD sheath (TC1 and TC2) were used for temperature measurements at various points in the system. The thermocouples were mounted as shown in Appendix IV.

The shell water was heated by a commercial steam - water heat exchanger (SHX). The exchanger was sized to raise the temperature from 60 to 180°F for a water flow of 80 gallons per minute (US) using 80 psi steam. This flow rate was more than twice the rate actually attainable through the shell side of the test heat exchanger. The exchanger was over-sized purposely in an effort to minimize any changes which were likely to occur in the shell inlet temperature when the shell stream was flow forced.

The temperature control valve on the steam (FCS) was equipped with an equal percentage plug and an electro-pneumatic positioner (FCSA). The control valve was sized on the basis of the heat exchanger specifications.

Two steam traps were used for condensate return. The main trap, STA, passed condensate into an overheat condensate return

line. However, if the pressure in the jacket of the heat exchanger, SHX, was insufficient to lift the condensate into the return header, then trap STA would flood and the smaller secondary trap, STB, which is positioned with its inlet four inches above that of trap STA, passed the condensate to drain. Both traps which were the float and thermostatic type, were chosen because of the wide range of steam pressures over which they were designed to operate.

5.3 Real-Time Use of the IBM 1800 Data Acquisition and Control System

5.3.1 General

A data acquisition and control system (DACS) such as the IBM 1800 used in the work for this thesis, has the basic features which make it possible to monitor and manipulate process variables in almost any fashion desired. The limiting factor for any particular implementation is the existing software which may require changes that are simple or extremely complex. The author's approach to the use of the DACS for this work evolved from the features that were available on the system in late 1968. The resulting programs will not necessarily be the most efficient for use on other systems, even other IBM 1800 systems.

The DACS was programmed to allow pulse test experiments to be conducted entirely from within the digital computer laboratory. At some time during an experimental run the DACS was able to perform the following basic operations:

a. control process variables to a set point

b. calculate an on-line heat balance

c. control and manipulate a flow valve for pulse tests

d. acquire and store data from the process at a rapid sample rate

The control to a setpoint utilized the standard direct digital control (DDC) software package supplied with the system. The heat balance and the execution of pulse tests was achieved with FORTRAN programs written by the author while the rapid data acquisition was made possible by a sub-routine written by the DACS Centre staff.

5.3.2 Control and Data Acquisition

The department's IBM 1800 DACS used the IBM DDC package with some modifications which were introduced by the DACS Centre staff. A general description of the features of this package are provided here in an effort to simplify the explanation of the organization of the DACS - heat exchanger interface.

Figure 4 is a schematic diagram showing the various elements which would constitute a typical control loop for the DACS, including both the hardware and software. Hardware components which were electrically connected and were physically in close proximity are shown connected by straight lines. Those which were electrically connected but were physically separated, are joined by dotted lines. The process operators console (POC) and the sample and hold amplifiers, called Current Output Stations (COS), were not electrically connected but were in close proximity physically in order to simplify operation of the process. The software, which permanently resided in the core memory of the DACS, was divided into four categories represented by

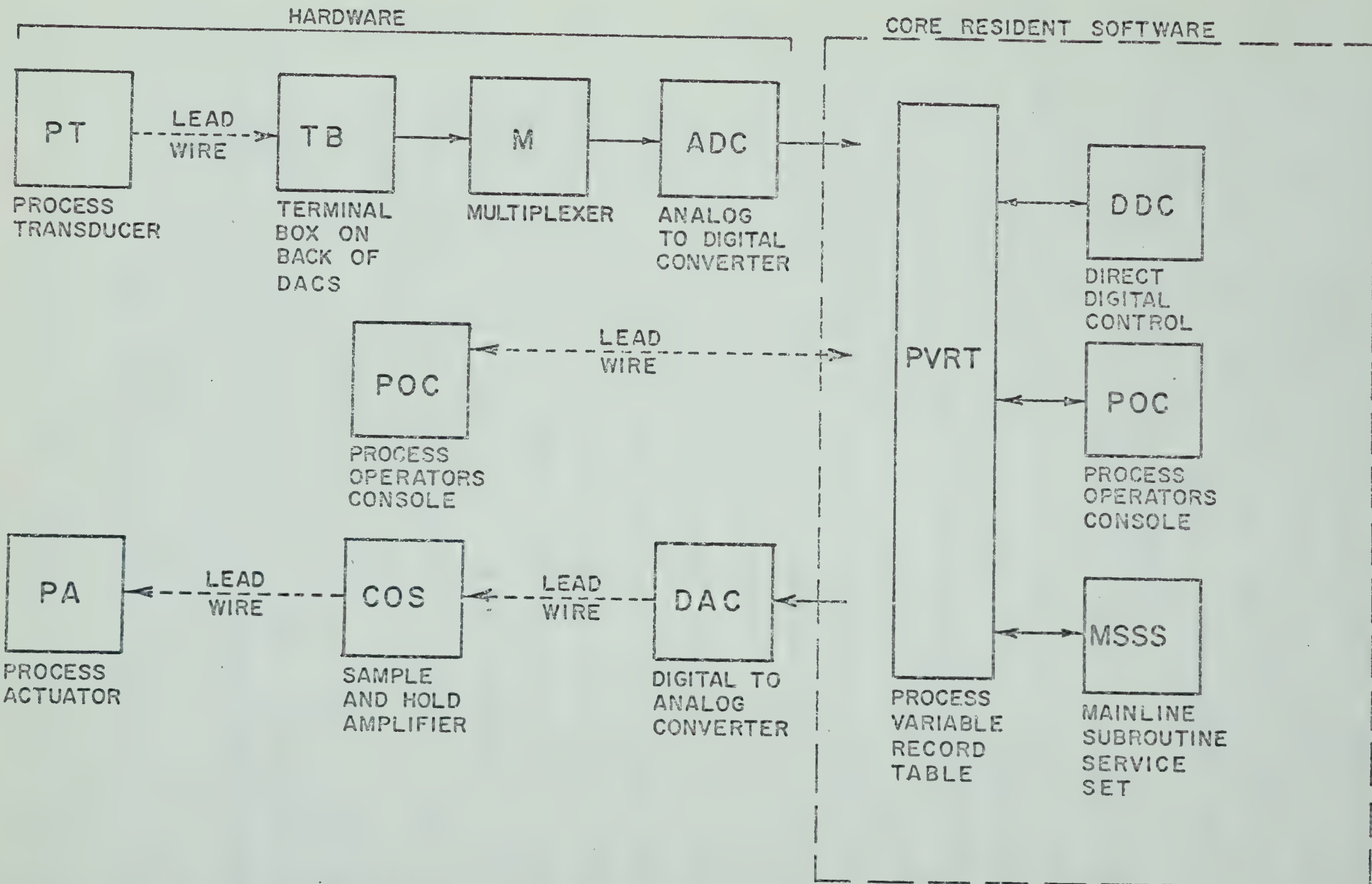


FIGURE 4 - SCHEMATIC DIAGRAM OF A DDC LOOP

the four smaller boxes inside the large dashed line box.

5.3.2.1 Hardware in a Control Loop

In DDC control the function of the hardware is to provide a means whereby the software can communicate directly with the physical process. Figure 4 illustrates the components involved with the direction of information flow shown by arrows.

For the input, the process transducer signal (PT), generally a voltage, is transmitted via a lead wire from the process to a pair of terminals in the DACS terminal box. The signal remains at this point until the DDC software references the information. At the time of request, the multiplexer, (M), selects the terminal pair and permits the signal to be passed through to the analog to digital converter, (ADC), which changes it into a binary number acceptable to the computer.

For the output, a digital value, which has been calculated by the DDC software, must be passed to the process actuator, (PA). The first stage of the sequence involves a digital to analog converter, (DAC), which generates a suitable analog signal for the sample and hold amplifier called a current output station, (COS). The COS supplies to the PA a continuous milliampere signal that is equivalent to the latest value transmitted from the DAC. This value is updated with each cycle of the control calculations.

With the hardware configuration just described, the same M, ADC and DAC are shared by all the control loops while an individual PT, PA terminal board pair and COS are required for each loop. Using this configuration, it is possible for the DDC software to simultaneously maintain control of several loops by addressing each

respective input and output in order and performing the necessary control calculations.

5.3.2.2 Software in a Control Loop

The DDC package used by the DACS features a relatively simple technique for building control loops. Basically the parameters which specify the control loop are punched onto cards (up to three may be required) in a coded form and the cards are then read into the DACS via the card reader. The information coded is divided into words, each of which, in part or in whole, specify such information as the address of the input or output, measurement, setpoint, alarm limits, alarm status, type of control action, engineering units, etc. The information for all of the control loops which have been specified is stored in the core memory of the computer in an area called the process variable record table (PVRT). For a given loop, some of the words will be changed every time the control calculations are made (e.g., measurement, error, output), other words will generally be changed only by the operator (e.g., setpoint, alarm limits, control constants), while still others will rarely be changed (e.g., address for measurement or output, loop identification on number).

The types of loops which could be formed may be divided into three general categories:

a. control

b. data acquisition

c. ring buffers

The first type is more or less self-explanatory; the measurement was read, then the desired output was calculated and transmitted to the process. The second type was used when the value of a process variable was required. In this case a standard control loop was formed but the parameters associated with the type of control and the output were not specified. The third type made it possible to obtain the time history of the measurement, setpoint, error or output from some other loop. Basically, the variable of interest was read and stored in a buffer each time the control calculations updated the value.

The principle for the operation of the DDC software can best be explained by reference to Figure 4. Here the PVRT, which contained the up-to-date information for each loop, supplied information to and received information from three sources. The first, DDC, updated the PVRT continually and automatically by scanning the parameters in each loop and performing the necessary operations at specifed time intervals. The cycle time was measured in seconds and must be a power of two with zero as the minimum power. The second, the process operator's console (POC), provided the process operator with the facility to read or change information in the PVRT. Using the hardware of the POC and its associated software, the process operator can read any word in any loop and can change the value of all but a few. The POC hardware consisted of an input/output (I/O) typewriter and a series of push-buttons. The values of words requested were typed on the typewriter by the DACS and the values of the words changed were entered by the typewriter. The third, Mainline Subroutine Service Set (MSSS), provided a set of Fortran called subroutines, which may be used to read or change information in the PVRT through Fortran programs resident in the DACS computer. The author used these subroutines extensively in the programs

which calculated the on-line heat balance and implemented a pulse test.

5.3.2.3 DACS Heat Exchanger Interface

Figure 5, which uses the same symbolism for heat exchanger hardware as Figure 3, illustrates how the DACS was interfaced with the heat exchanger system. The small rectangular blocks in Figure 5 are used to represent each DACS function and the dotted lines to represent connecting electrical wires between the heat exchanger system and the DACS. The arrows on the dotted lines indicate the direction of information flow. The symbol designations inside the small rectangular blocks signify the DACS functions as follows:

a. DDC - Direct Digital Control

   Utilizes the features of the standard DDC package

b. DAS - Data Acquisition Slow

   Utilizes the features of the standard DDC package which permits sampling at a maximum rate of one point per second. (Data read in this way can be saved in ring buffers.)

c. DAR - Data Acquisition Rapid

   Utilizes the special software feature developed by the DACS Centre staff. (Permitted sampling at a rate of forty points per second.)

The subscripts, i, o, h and c, beside some of the hardware symbols in Figure 5 refer to in, out, hot and cold respectively. In the following material, the system organization and its operation is discussed by reference to Figure 5.

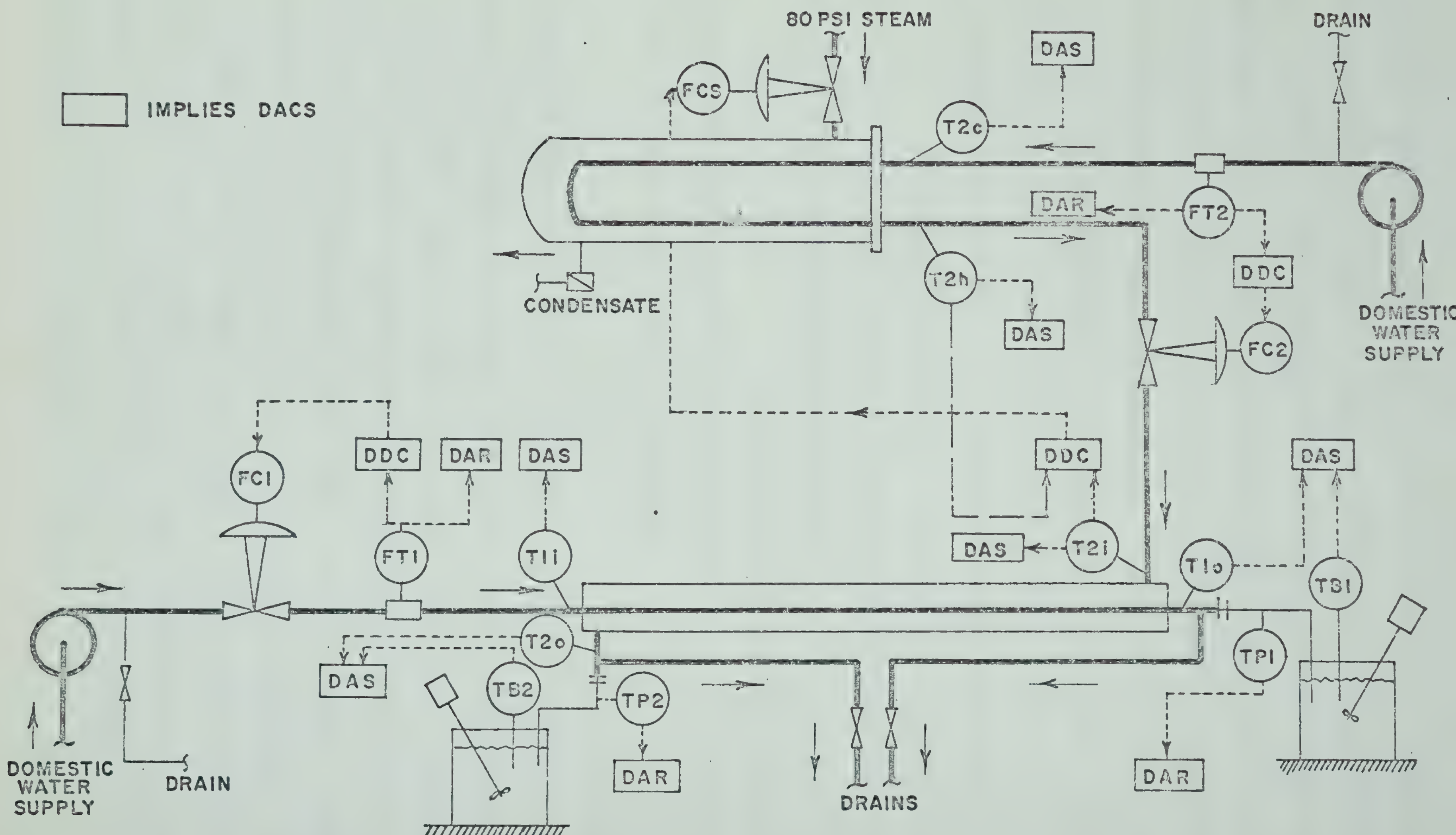


FIGURE 5 - SCHEMATIC DIAGRAM HEAT EXCHANGER SYSTEM
(OPERATIONAL)

The tube flow, the shell flow and the shell inlet temperature on the test heat exchanger were the only variables controlled by the DACS. The tube inlet temperature, which was that of the domestic water supply, remained constant during any series of tests. The controlled variables employed the features of the DDC package and control loops were formed using the integral plus proportional algorithm on a one second scan cycle. This control was satisfactory so no particular effort was devoted to trying the other control algorithms available. Also, since the load on the DACS produced by the three loops was negligible, no particular effort was devoted to the selection of an optimum scan cycle.

The flow measurements were obtained from turbine meters FT1 and FT 2 and the control action was implemented through the respective control valves FC1 and FC2.

One of two possible thermocouples was used in the control of the shell inlet temperature. The selection was made by using the POC. The location of these thermocouples was:

a. The outlet of the commercial shell and tube heat exchanger, SHX.

b. The actual inlet to the shell of the test heat exchanger, THX.

In the latter case a transport delay of approximately 15 feet of pipe existed but it offered the advantage of directly controlling the inlet temperature to the shell of THX at a specified value. Experience revealed that the detuning necessary to control from this measurement point was not a serious problem for operation near steady state. As a result, start-up was based on control from the first point with steady state operation using the second. No flow forcing was used in the

shell, so large load changes were unlikely after steady state was attained on the shell inlet temperature. The option for controlling the shell stream temperature from the outlet of heat exchanger SHX is shown as a dashed line in Figure 5.

The slow data acquisition, indicated by DAS in the rectangles on Figure 5, made use of the data acquisition feature of the DDC package as explained in Section 5.3.2.2. The data acquisition loops formed operated on a one second scan cycle, again because the load on the DACS at the time was light and there was no need to select a more optimal value. Generally the data acquisition loops, which were used, served two purposes. First, the loops provided a ready means for the operator to obtain up-to-date values of process variables by using the POC features. This was found to be particularly useful during the check out of the components on the heat exchange system. Second, the loops provided up-to-date values for on line programs running in the computer. In particular, the slow data acquisition loops reading the inlet and outlet temperatures of the streams on the test heat exchanger were referenced extensively by the on-line heat balance program through the facilities of the MSSS.

Ring buffer loops were used for plotting the measurements of the loops under control when the control constants and digital filter constants were being selected. Also, they were used for graphical study in the development of the pulsing program. Four ring buffers stored the response of the measurement, setpoint, error and output of the tube flow when a pulse was introduced.

The rapid data acquisition indicated by DAR in the rectangular blocks on Figure 5 was achieved through a special Fortran accessible subroutine written by the DACS Centre staff. The subroutine, as used, was restricted to a rate of forty samples per second on two variables for a time duration of forty seconds. However, the DACS Centre staff are developing a more generalized version of this subroutine. Basically, the subroutine addressed two multiplexer points alternately, read the binary equivalent of the voltage values present and stored these binary values in a file on one of the disk storage units. When the rapid data acquisition was completed, the data was sorted, so that the values from each of the multiplexer points was placed in a separate disk file in chronological order. This was achieved through the use of a sorting subroutine also written by the DACS Centre staff. The program, STDAT, listed in Appendix III, was the mainline which used this subroutine. The values in the files after sorting were simply binary coded integer numbers which had little physical significance until converted into engineering units. The procedure used for this conversion is covered in Section 5.4.2.1.

For the results reported in this work, the tube flow was the forcing function or the input pulse and the shell outlet temperature was the response or output pulse, therefore, the flow turbine FT1 and the thermopile TP1 were the transducers which supplied the signals recorded by the rapid data acquisition subroutine. It was possible for the DDC and rapid data acquisition of the tube flow to take place simultaneously using the same transducer and multiplexer

point. As shown by Figure 5, the heat exchanger - DACS interface was such that shell flow and the tube outlet temperature could have been monitored through the use of the rapid data acquisition subroutine, although time limitations did not allow this case to be considered in this work.

5.4 Experimental Procedure

As mentioned in Section 5.3.1, the DACS was programmed to allow pulse test experiments to be run entirely from within the Digital Control Laboratory. The experimental procedure used can be divided into two parts; that which took place in real time and that which did not, or that which involved the acquisition of experimental data and that which involved the analysis of same. The experimental procedure has been divided into these two parts for discussions in the following two sections.

5.4.1 Experimental Procedure for Data Acquisition

The experimental procedure used for data acquisition relied heavily on features available on the DACS. To simplify the text in this section, it will be assumed that the reader is familiar with the information in Section 5.3. Programs which are referred to by a name in upper case letters are listed in Appendix III. References to the computer in this section imply the digital computing facilities of the DACS and references to writing out or entering information imply the use of the POC typewriter unless otherwise specified.

5.4.1.1 Heat Balance Calculation

The first program executed during an experimental run was the on-line heat balance Fortran program named WLHBL. Basically, this program utilized the MSSS features, along with slow data acquisition loops, to obtain current values for the flows and temperatures on the streams of the test heat exchanger. From this information the heat transport ratio was calculated and written out along with averages of the values measured from the process. To execute WLHBL, it was necessary to bring the program from disk storage into the computer core. This was accomplished by pressing the appropriate button on the POC and entering the number 9. When WLHBL was present in core, two time parameters had to be entered before execution would commence. The first parameter called ISAMPLE PERIOD set the time (seconds) which would elapse between each set of readings that were obtained from the PVRT through the MSSS. The minimum value set for ISAMPLE PERIOD was one because this was equivalent to the rate at which the DDC package updated the required measurements in the PVRT. The values obtained each ISAMPLE PERIOD were saved in buffers until the heat balance calculations were made. The second parameter, ITOTAL TIME, set the time (seconds) which would elapse between heat balance calculations. Every ITOTAL TIME seconds the program would average the buffered values, calculate the heat balance using the averages and write out the results. For this program to function properly, ISAMPLE PERIOD had to be less than, or equal to, ITOTAL TIME.

Once the heat balance program had commenced execution, it would periodically report the heat balance results until

the operator would abort it by turning on data switches 7 or 8. If 7 was on, the run would terminate with the heat balance. If 8 was on, the pulsing program was brought from disk storage into the computer core for execution. In this way it was possible for the operator to use the heat balance as a guide in determining the necessary steady state before conducting a pulse test. Generally, a sample period of three seconds and a total time of sixty seconds was used and at least three satisfactory heat balances were obtained before the pulse test was conducted.

5.4.1.2 Introduction of Flow Pulse

The program PULSE, which was written in Fortran, controlled each pulse test. Generally speaking, PULSE started the rapid data acquisition and after a brief period at steady state, utilized the features of the MSSS to introduce an input pulse by manipulating the control valve on the tube flow of the test heat exchanger. Two methods for manipulating the control valve were provided. The first pulsed the flow by changing the setpoint in the associated DDC loop; the second placed the loop in the manual mode immediately before the pulse and then manipulated the valve by changing the output of the loop. The loop was then returned to the automatic mode following closure of the pulse. Preliminary investigation revealed the second method to be more satisfactory than the first, so it was used for all tests reported in this thesis.

The listing for PULSE gives a reasonably comprehensive description of the parameters, which had to be entered

before the program executed. ISASS and ISPUL need no comment; however, CHGTYP, CHGL1 and CHGL2 require further explanation. CHGL1 can be considered as the change limit for the beginning of the pulse. The relative absolute magnitude of CHGTYP and CHGL1 determined whether the beginning of the pulse was a ramp or a step. If CHGTYP was less than CHGL1, the begining was a step, but if CHGTYP was greater than CHGL1, the begining was a ramp with a slope of CHGL1 units per second. For the end of the pulse the same idea applied with CHGL2 taking the place of CHGL1. If the output was being decreased, the signs of CHGTYP, CHGL1 and CHGL2 had to be negative; if it was being increased, they had to be positive. With this technique it was possible to generate four basic pulse shapes; rectangle, triangle, ramp and reversed ramp. Only the rectangle and the ramp were used for the results reported herein. The parameter PLVAL permitted the specification of the maximum deviation that the pulse could have from steady state, so the system tested could be protected from saturation. This meant a ramp pulse could be truncated if the amplitude exceeded PLVAL before the pulse duration had elapsed. The parameter MANUL requires no further explanation.

When the above parameters had been entered, the program controlled the pulse test by completing the following steps in sequence:

a. When data switch 11 was on, four ring buffer loops were made operable. (From these buffers the time history of the measurement setpoint, error and output could be plotted for the loop pulsed. These plots were particularly useful for selecting

values for the parameters which determined the pulse shape. Data switch 10 could be used to by-pass the rapid data acquisition program during such studies.)

b. The voltage generated by the reference thermocouple (located in the thermopile reference bath) was read once a second for five seconds along with two variables in the DACS which are used for calculating cold junction compensation. The values obtained were averaged, the cold junction compensation voltage was then added to that generated by the thermocouple and the result written out.

c. The rapid data acquisition program was initialized so that it idled ready to start reading data immediately after specified indicator was set to a non zero value.

d. The time of day was recorded from the system clock and the time when the pulse was to begin was calculated.

e. The MSSS was used to obtain the current value of the output for the loop to be pulsed. This value and the value of the output to occur at the peak of the pulse were written out. (As discussed earlier, the setpoint could have been used.)

f. The data acquisition indicator was set to a non zero value and the data acquisition was started immediately.

g. When the time to introduce the pulse was reached, the loop to be pulsed was placed in the manual mode. The time when the pulse was to be deleted was calculated and the output to the valve was changed as required to generate the pulse shape

specified by the parameters entered. Each new value of the output was placed in the PVRT through the facilities of the MSSS.

h. When the time to delete the pulse was reached, the output was returned to the original value and after three seconds the loop was placed back in the automatic mode.

i. If data switch 13 was turned on before the run was completed then the program would short at this stage in the sequence of steps.

j. When rapid data acquisition was completed (forty seconds after start), the second step in the sequence was repeated and the values obtained at the beginning and the end of the run were averaged and the result was written out.

k. The program STDAT which sorts the acquired data was automatically brought from disk storage into core for execution. Parameters were entered to specify the numbers of the multiplexers from which the data was read and the numbers of the disk files in which the sorted data was to be stored. When STDAT had completed execution, the experimental procedure for data acquisition was complete.

The experimental pulse tests used to determine the transfer function relating tube flow to shell outlet temperature were restricted as follows:

a. The steady state tube and shell flows were eight and fifteen U.S. gallons per minute, respectively, and the shell inlet temperature was controlled to 150°F.

b. Five seconds of data was taken at steady before the pulse and flow pulses were generated by manipulating the output signal to the tube flow control valve.

c. Two basic pulse shapes were used, the rectangle and the ramp. In both cases the maximum change in the output was chosen so the tube flow pulse would have a peak of approximately twelve U.S. gallons per minute.

d. Pulse durations of 1, 3, 6, 10, and 15 seconds were used for the square pulse while 6, 10, 15 and 20 seconds were used for the ramp.

5.4.2 Experimental Procedure for Data Analysis

The data analysis was accomplished using a series of Fortran programs written to utilize the data processing capabilities of the DACS. The listings for these programs can be found in Appendix III. References to the computer in this section imply the digital computing facilities of the DACS and references to writing out or entering information imply the use of the POC typewriter unless otherwise specified.

The data analysis programs were stored on the disk storage units and all data they processed were stored in disk files. To simplify the explanation of the use of the data analysis programs, it is necessary for the reader to understand the basic structure of the disk files. A total of eight files were accessible to any of the programs which processed data. Each file was 5,120 words long which provided enough room to store 5,120 integer values or 2,560 real values. By the design of the DACS, each file had to be divided into

equal sized records which were used in the disk read - write operations. Basically, the read - write statements specified the file number and the number of the record accessed in that file. A full record of data had to be transferred by each read - write operation. For the programs in Appendix III, a record size of eighty words was used.

Because both the programs and the data were stored on disk, it was possible to control a complete data analysis by entering instructions through the POC typewriter. To begin the data analysis, the initialization program, SET, had to be executed through a single card read operation. The function of SET was to initialize an indicator for the executive program CHIEF and to write out a list of the data analysis programs along with an associated number. CHIEF, which was automatically executed immediately after SET, provided a ready means for the operator to select the order in which the data analysis programs would be executed. This was done by entering the associated numbers in the desired sequence. When the last program specified in the sequence had completed execution, a completely new sequence could be entered. The programming was done so that there were absolutely no restrictions on the sequence of execution of the programs; however, the purpose of the analysis introduced certain logical restrictions. The flow chart in Figure 6 illustrates the most common sequence used by the author. The decisions indicated were those made by the operator when determining the desired sequence.

The parameters which had to be entered by the operator before each program would execute are described in the appropriate

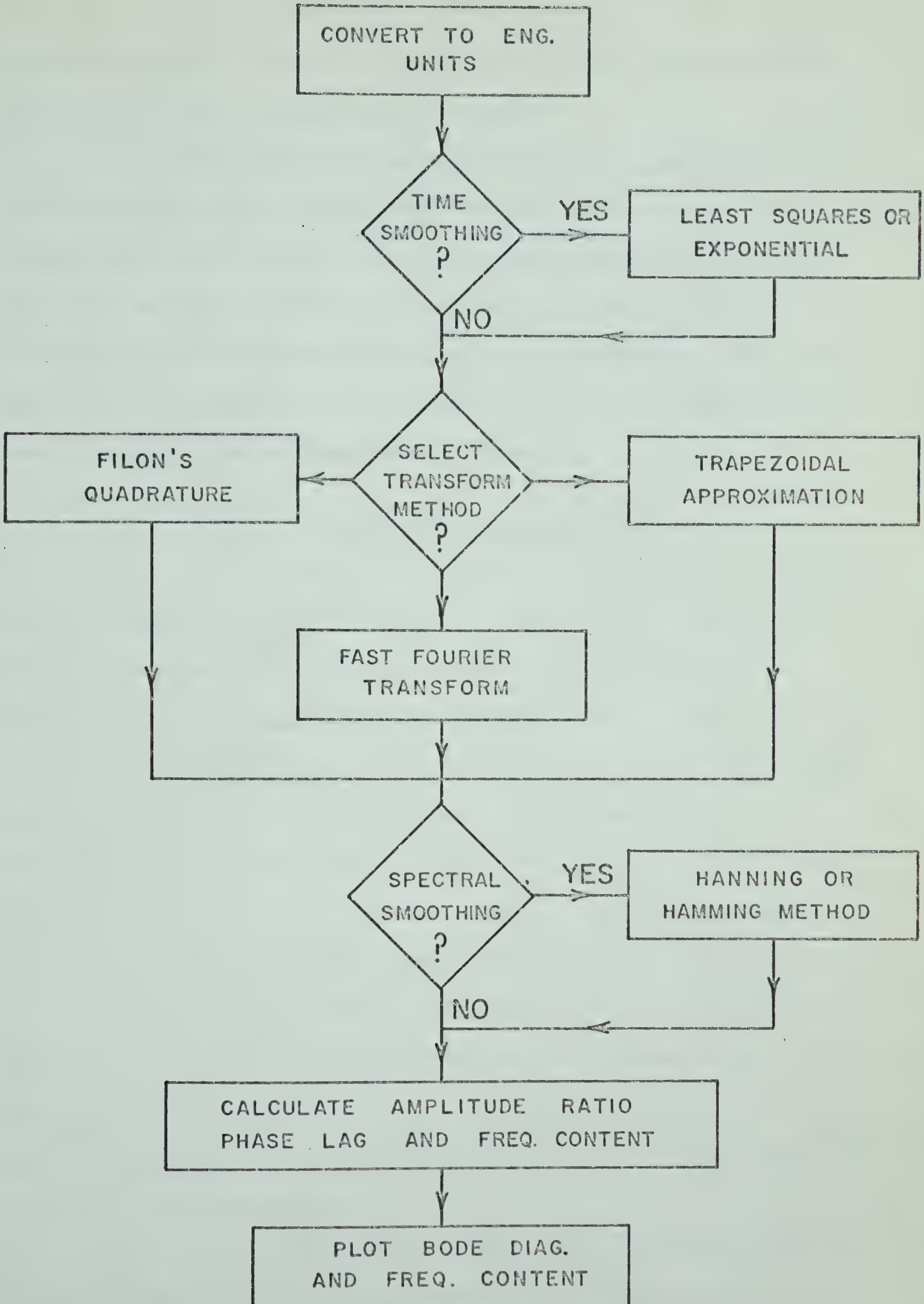


FIGURE 6 - DATA ANALYSIS PROGRAM EXECUTION SEQUENCE

listing in Appendix III, and they are not mentioned in the following text unless further explanation is required.

The listings of the subroutines used by each program are included in Appendix III immediately following the listing of that program. Listings of DACS centre subroutines used are not provided, since they apply specifically to the DACS used in this work and, therefore, are not of general interest. In the following text, each data analysis program is described briefly in the order they would be used to complete the sequence shown in Figure 6.

5.4.2.1 Conversion to Engineering Units

When the pulse data for the tube flow and shell outlet temperature had been sorted and stored on disk, each value was in the form of an integer number which had little significance. (Section 5.3.2.3). These values were derived by the ADC from the voltage signal on each pair of terminals to which flow and temperature signals were supplied. The voltage signal was linearly converted in terms of the range on the terminal pair relative to the binary range of -32768 to +32767, i.e., on a -5 to + 5 volt range, a +3 volt signal would be stored as $(\frac{3}{5})$ (32767) = 19660. The first step in any data analysis was to convert the integer data into engineering units. Two programs were used to accomplish this objective, one for flow data named VTFLO, and one for temperature data named MVTMP. Generally speaking, both programs converted the binary numbers back to volts and then converted the volts to engineering units.

The program VTFLO used a linear conversion from binary to volts, a least squares polynomial for conversion from volts

to frequency in Hz (Section 5.2.2) and a linear conversion from frequency to U.S. gallons per minute.

The program MVTMP was more complicated in that the thermopile millivoltage had to be divided by the number of probes and added to the reference thermocouple voltage before it could be converted to temperature. This situation was even further complicated by the failure of the original thermopile. (See Section 5.2.2 and the general comments in the program listing for MVTMP.) The steps which were involved in converting the temperature were as follows:

a. The reference thermocouple voltage supplied during the execution of PULSE (Section 5.4.1.2) was converted to the equivalent temperature (T) in degrees Fahrenheit through the use of a least squares polynomial fit to the chromel constantan thermocouple table.

b. The voltage (Y) which would be generated by a copper - constantan thermocouple measuring temperature T was calculated using a least squares polynomial fit to the copper - constantan thermocouple tables.

(Note: The above two steps are necessary because of the failure of the original thermopile and the use of a copper - constantan substitute - Section 5.2.2)

c. In a Fortran DO loop, each binary number is converted to volts, divided by the number of thermopile probes and added to Y to obtain the copper - constantan voltage (X) at the thermopile hot junctions. Then X is converted to degrees Fahrenheit

through a least squares polynomial fit to the copper - constantan thermocouple tables. (Note: For further information on the least squares polynomial fit, see Section 5.4.2.7.)

When programs VTFLO and MVTMP had completed execution, it was general practice to dump the flow and temperature data from the disk files onto cards. This was accomplished using a standard DACS function which dumped data from files onto cards in binary form. The practice was used to ensure that the results from any experimental run would be preserved because it was possible for the operator to inadvertently enter parameters which would cause a program to write over the experimental data. If this occurred, the data could be restored from the cards using a standard DACS storing function. Also, this practice ensured that analysis of any run could be repeated if questions arose at a later date. A total file of 5,120 words could be stored on a deck of cards approximately one inch thick.

5.4.2.2 Time Domain Smoothing

Neither analog nor digital smoothing was applied to the pulse test data obtained, so if time smoothing was required, it had to be applied during the data analysis. The program SMOTH was written to provide a series of smoothing algorithms which could be applied to the data, if required. This program also contained algorithms for frequency domain smoothing; however, only the time smoothing is considered in this section.

In general, the method of smoothing was by convolution using symmetric weighting functions (69) so an equal number

of points were considered on each side of the point being smoothed. Two methods of time smoothing were available, exponential and least squares. The first, which was developed by the author, applies exponentially decreasing weighting functions to raw data points as they occur further and further away from the point being smoothed. A parameter (B), which must be less than 1.0, determined the degree of smoothing to be applied. For example, if B was 0.8, the raw value being smoothed (R) would contribute 0.8 of its value towards the smoothed value (S), the raw values on each side of R would contribute 0.64, the raw values two away from R would contribute 0.4096 and so on until the total number of points involved in the smoothing calculation have been utilized. The smoothed value S was then normalized by dividing by the sum of the weighting functions. Three, five or seven points could be used for exponential smoothing depending upon the value entered for the parameter NS. The second method used smoothing by a least squares procedure developed by Savitzky and Golay (69). Basically, the method involves applying a series of weighting functions to the raw data in the fashion illustrated in the example above. The difference is that the weighting functions are derived to provide the exact equivalent to a least squares polynomial fit to the data points considered. The fit then determines the smoothed value of the central data point. In the program SMOTH options were available to provide smoothing of five, seven, nine or eleven raw points using second and third degree polynomials and seven, nine, eleven or thirteen raw points using fourth and fifth degree polynomials.

Preliminary examination of the experimental data revealed that the noise was of a sufficiently low magnitude so as to make the effects of time domain smoothing negligible in the final frequency response diagrams derived. For this reason only the case of a third order polynomial fit using eleven points was used.

The spectral smoothing or frequency domain smoothing using SMOTH is covered in Section 5.4.2.4.

5.4.2.3 Fourier Transform

Two programs KLTLY and FREQ were written to calculate the Fourier transform. In both cases the procedure involved calling a subroutine to calculate the actual transform, first for the output and then for the input pulse.

Program KLTLY used the Fast Fourier Transform (FFT) technique which was expressly developed to provide a fast, efficient means of digitally evaluating the Fourier transform from discrete time data. Three subroutines were used by KLTLY, namely INITL, TNFRM and REVER. The function of INITL was to initialize the pulse data for the FFT by averaging IRASS records of data at the beginning of the pulse test data and subtracting this value from the remaining points to obtain a pulse curve departing from a zero baseline. INITL also wrote out the error of closure for every third point of the last forty values used in the transform calculations and calculated the area under the input pulse when the input transform was being computed. The subroutines TNFRM and REVER conducted the actual calculation of the FFT. The card decks for TNFRM and REVER were supplied to the author by J. H. Smith,

of Trinity University Computer Centre, Trinity University, San Antonio, Texas. The subroutines were written for a DACS similar to the one used for this work, so no adaptation was necessary on implementation. An explanation of the use of these subroutines precedes the listing of TNFRM. Because the FFT operated satisfactorily, no detailed study was made into its derivation; however, some general information is given in Section 3.2.4.

The program FREQ provided the facility to calculate the Fourier Transform by either the trapezoidal or Filon's (27) quadrature using subroutines ABCD1 or ABCD2, respectively. A total of six subroutines were used by FREQ; they were REDPT, ABCD1, ABCD2, SUMS, ARPHF and ANGLE. In the following text, the function of the first four subroutines is discussed in the order in which they are listed above. The last two will be discussed in Section 5.4.2.5.

Subroutine REDPT performed the function of reducing the number of time data points used in the Fourier Transform calculation. Basically, the subroutine scanned the original data selecting a point every NPICK points and placed the selected values in a new disk file. The effect of REDPT on the data is discussed in Section 6.3.2.1. For most analyses conducted using the program FREQ, subroutine REDPT was employed because computation times were excessive (in the order of twenty minutes) for calculating more than about forty frequency points from 360 time points. In general, the computing time is directly proportional to the product of the number of frequency and the number of time points involved.

The subroutine ABCD1 used the trapezoidal method of quadrature to evaluate the A, B, C and D trigonometric integrals required to calculate Fourier transform as shown in Section 3.2.3. The following steps were completed in sequence during the execution of ABCD1:

a. The number of frequency points to be calculated was determined from the parameters FINT, FINC and FLST. The number was selected such that the frequency points begin at FINT and increment by FINC up to a value greater than or equal to, FLST depending upon the maximum frequency required to calculate an integral number of full records, i.e., the number of frequency points was always a multiple of forty. Full records were calculated because the disk read - write operation always transferred full records of data and was deemed useful to not have part of a record filled with meaningless values, since this could complicate the writing of other programs.

b. The steady state value of the process before the pulse occurred was evaluated by averaging IRASS records of steady state data. IRASS was a parameter entered by the operator.

c. The transcendental integrals were evaluated for each frequency point required using the trapezoidal quadrature. In the calculation procedure the steady state value was subtracted from each time data point to yield a closed pulse which deviated from a zero baseline. (Note: in a properly executed pulse test, the values of the end points are zero, so, for

convenience in programming, the trapezoidal quadrature used in ABCD1 neglected these points.)

d. The error of closure was written out for every third time point of the last forty involved in the transform calculations. An error of closure existed if the response variable did not return to the original steady state.

The subroutine ABCD2 used Filon's quadrature (27) to evaluate the A, B, C and D trigonometric integrals associated with the Fourier Transform. The sequence of steps completed during the execution of ABCD2 was identical to those in ABCD1 except for the quadrature formula used.

The subroutine SUMS used Simpson's rule to calculate the area under the input pulse.

5.4.2.4. Spectral Smoothing

As mentioned in Section 5.4.2.2, the program SMOTH contained algorithms suitable for frequency domain or spectral smoothing as well as the algorithms for time domain smoothing. The algorithms used for spectral smoothing were obtained from Blackman and Tukey (5) and Blackman (4). Three choices were available for spectral smoothing:

a. the Hanning method

b. the Hamming method

c. the combination method which involved first smoothing by Hanning then smoothing those results with a third algorithm. This method is identified as 'HANNING SECOND PASS' in the listing of SMOTH.

Further details concerning spectral smoothing were outlined in Section 3.4.

5.4.2.5. Calculation of Amplitude Ratio Phase Lag and Frequency Content

When the Fourier Transform had been evaluated, it was necessary to combine the results to obtain the amplitude ratio and phase lag for the process and the frequency content of the input pulse. In the case of program FREQ, the subroutine ARPHF performed the necessary calculations from the results of subroutines ABCD1 or ABCD2. In the case of program KLTY, the program BFKTY was used.

The procedure for calling ARPHF dependent upon the use of spectral smoothing. If spectral smoothing was not used then program FREQ called subroutine ARPHF directly, however, if spectral smoothing was applied then program BUCWS was executed. The function of BUCWS was simply to provide a mainline program which the operator could use to execute subroutine ARPHF separately. The calculations involved in ARPHF were the simple arithmetic operations given by equations (3.13) through (3.21), except for the arctangent function (Equation 3.16) which required the subroutine ANGLE that was called by the subroutine ARPHF.

The form of ANGLE used in this work was not totally satisfactory because it was prone to introduce discontinuities in the phase lag graphs when there was a significant amount of noise in the Fourier Transform values. To evaluate the arctangent ANGLE used, the FORTRAN function subprogram ATAN which always calculated an angle lying between $\pm$ 90 degrees. Since the phase lag for most systems goes beyond -90 degrees, it was necessary to provide a memory function and

an adjustment of $\pm$ nx 180 degrees (n = 0, 1, 2, - - -), so that an accumulative angle could be computed from successive values of the real and imaginary components of the transfer function. The technique worked satisfactorily so long as the difference between the values of successive components did not indicate angle change in the order of 90 to 180 degrees, because if this occurred, the memory function would lose track of its position making the information obtained from ANGLE meaningless.

The occurrence of a breakdown in ANGLE was easily detected from the graphs since sharp discontinuities of 180 degrees or more appeared. These discontinuities commonly occurred at the zeros of the frequency content curve, or whenever an error of closure in the pulse introduced sufficient noise in the values of the Fourier transform.

Phase lag graphs presented in this work do not include points beyond the frequency at which there was an obvious breakdown in subroutine ANGLE. However, the existence of a breakdown can be determined by comparing the number of points plotted in corresponding amplitude ratio and phase lag graphs. When the number on the phase lag graph is less, a breakdown occurred approximately at the frequency where the curve on the phase lag graph terminates.

The operation of program BFKTY was identical to that of subroutine ARPHF except that the parameters entered and the variable names were made compatible with those in program KLTLY. The subroutine ANGLE was used by BFKTY in the same way it was used by ARPHF.

5.4.2.6 Plotting Frequency Response Diagram

All the graphs presented in this work were obtained from the DACS Centre Calcamp Plotter using the generalized plotting subroutine NWPLT which was written by fellow graduate student, R. B. Newell. The program PWRSL was written by the author to utilize subroutine NWPLT which in turn used subroutines NWLNS, NWMAX, NWLGS, LGRID and NWANN. A listing of each of these subroutines can be found in Appendix III, following the listing of program PWRSL.

5.4.2.7 Computer Programs

Two other programs which were executable from the keyboard were PICK and WRITE. The first was simply a mainline program to allow the operator to have direct access to subroutine REDPT (Section 5.4.2.3). The second was a utility program to allow the operator to manipulate data in the disk files. It had the facility to read integer or real data from a file and write into another file, punch it onto cards in E FORMAT or write it on the line printer in E FORMAT. The operator could specify the number of records that would be read from the file.

The program LINEA, which was used to calculate a least squares polynomial fit to tabular data (Sections 5.2.2 and 5.4.2.1), was written by fellow graduate student, V. Krishna. LINEA was programmed to run at the University of Alberta Computer Centre, so it was not run at the DACS Centre.

VI. DISCUSSION OF RESULTS AND CONCLUSIONS

6.1 Introduction

The results of both the theoretical and experimental analyses are presented in this chapter. All information appears in graphic form because the use of program NWPLT made it possible to obtain labelled graphs directly from data files. The labelling code which was used on the axes of each graph is explained in the following text. The independent variable for the data presented on each graph is time or frequency. The units of these variables are seconds and radians per second, respectively.

The label on the independent variable axis (abscissa) of the graphs is of the following general format:

MIN VALUE	AXIS UNITS	MAX VALUE
	Pmnxxy_ijkl_abc_efgh	

The first line of information consists of:

a. The minimum and maximum values which coincide with "tick" marks at the end points of the axis.

b. One of two types of AXIS UNITS:

   1. LOG CYCLES where the scale is logrithmic and the ordinate axis intersects the abscissa axis at the one cycle per second point.

   2. (0.XXXE_XX_INTERVALS) where the scale is linear, X is a numerical digit and E implies exponentiation to the base 10. The value presented is equivalent to the difference between the "tick" marks along the axis.

The second line provides information on the graph identification, the source of the data, the type of system tested and the variable name as explained below:

a. Pmnxxy is the graph identification where the characters have the following significance:

P - Appears on all graphs.

mn - An alpha code associated with the type of data; the following cases occur;

TH - THeoretical time data

EX - EXperimental time data

FT - frequency data derived using the FasT Fourier transform

TP - frequency data derived using the TraPezoidal quadrature

FN - frequency data derived using the FiloN quadrature

xx - A numerical code associated with a particular run. A run is defined as any set of input and output time pulses which was obtained from a pulse test. Many different graphs of frequency data are derived from one run. Run numbers range from 01 to 08 inclusive for theoretical data and 10 to 20 inclusive for experimental data. Run 15 has not been included in the results.

y - An alpha character which may be any character in the alphabet.

Note: The xxy code has the following significance when comparisons are made within experimental or theoretical results.

- When "xx" is the same but "y" is different for two graphs, the graphs are different but have been derived from the same run.
- When "xx" is different but "y" is the same for two graphs, the graphs are similar but have been derived from data obtained on different runs.

b. ijkl is an abbreviation for the source of the data. The following cases occur:

- time data

  THRY - theoretical

  EXPR - experimental

- frequency data

  FAST - derived using the Fast Fourier transform

  TRAP - derived using the trapezoidal quadrature

  FILN - derived using the Filon quadrature

c. abc is a two part code related to the system tested as follows:

ab - Alpha numeric code associated with the type of system pulse tested.

1_ - Theoretical; first order system.

11 - Theoretical; cascaded first order systems.

2_ - Theoretical; second order system.

EX - Experimental; test heat exchanger.

c - Alpha code to indicate whether or not the time data has been reduced by subroutine.

REDPT (Section 5.4.2.3)

F - Data not reduced (original)

R - Data reduced

d. efgh represents the name of the independent variable plotted. The following two cases occur:

TIME - time

FREQ - frequency

The dependent variable axis on the graphs appears in the following general format:

pqr_stu_vwx_zzzz

MIN VALUE AXIS UNITS MAX VALUE

The first line of information provides the name of the dependent variable plotted, filtering used, if any, and the number of data points plotted as is explained below:

a. pqr_ stu represents the two words in the name of the dependent variable. The following names are used:

AMP. RATIO - amplitude ratio

PHASE LAG - phase lag

FREQ. CONT. - frequency content

INPUT PULSE - input pulse (theoretical)

OUT PULSE - output pulse (theoretical)

FLOW GPM US - flow gallons per minute U.S. (input pulse experimental)

TEMP DEG F - temperature degrees Fahrenheit (output pulse experimental)

b. vwx is an alpha numeric code associated with the type of filtering used. The following cases occur:

R - Raw data - no smoothing used.

RP311X and P311X - Polynomial least squares smoothing using a third order polynomial with eleven data points applied "X" times to the time data. When "X" is a blank, 1 is implied.

M1311X - Same as case RP311X and P311X, but with hamming spectral smoothing also applied once.

N2311X - Same as case RP311X and P311X, but with hanning spectral smoothing also applied twice using the appropriate algorithm each time.

M13 and M1 - Raw time data with hamming spectral smoothing applied once. (The 3 has no meaning.)

N23 and N2 - Raw time data with hanning spectral smoothing applied twice using the appropriate algorithm. (The 3 has no meaning.)

Note: For information on the filtering algorithms, see Sections 5.4.2.2 and 5.4.2.4.

c. zzzz are numerical values indicating the number of data points plotted.

The second line on the axis for the dependent variable has the same definition as the first line on the axis for the independent variable, as already discussed.

There are some graphs for which the axis labelling deviates slightly from the code above, however, these cases are explained in the section where such graphs appear. The general arrangement for the data places the input and output time pulses in appendices along with the frequency content graph for the input pulse. Theoretically generated data are located in Appendix I and the experimental data are located in Appendix II.

The amplitude ratio, phase lag and input pulse frequency content graphs, which differ from those in the appendices, are located in various sections throughout this chapter, where it was deemed most appropriate for making the desired comparisons. The exceptions to the above organization are theoretical runs 07 and 08. All the graphs related to these runs appear in Section 6.2.5.

6.2 Theoretical Analysis

The theoretical analysis used time domain pulse data generated using the technique outlined in Chapter IV. In this analysis, all input pulses were rectangular in shape and were restricted to a steady state value of six and a peak value of nine. These values were arbitrarily chosen because experimentation with a wide range of values resulted in no noticeable change in frequency domain results. A six and nine combination represents an input pulse magnitude of fifty percent of steady state which is reasonable for many physical processes. The theoretical pulse tests are numbered 01 to 08. Runs 01 to 06 deal with pulse testing a first order system, while run 07 is concerned with cascaded first order systems and run 08 with a second order system.

6.2.1 Effect of Input Pulse Duration

The results from runs 01 through 06, for a first order system with a five second time constant, demonstrate the effect of input pulse duration. Durations of 1.0, 2.5, 5.0, 10.0, 15.0, 25.0 and 75.0 seconds were used which are 1/5, 1/2, 1, 2, 3, 5 and 15 times the time constant of the system tested.

6.2.1.1 Effect of Input Pulse Duration on the Frequency Content of the Input Pulse

A comparison of the frequency content Graphs PFT01C, PFT02C through to PFT06C (Appendix I) demonstrates that the first zero of the frequency content curve of a rectangular pulse occurs at lower and lower frequency values as the duration of the pulse is increased. This trend agrees with the mathematical theory presented in Section 3.2.2. The curve on Graph PFT02C exhibits its first zero at a frequency of approximately 2.46 radians per second. It will be recalled from Section 3.2.2 that the frequency content curve for a rectangular pulse exhibits its first zero when the relationship $w \cdot d = 2\pi$ is satisfied. In the case of run 02, the value of "d" is 2.5 seconds so the first zero would occur at $w = 2.512$ radians per second. Considering the accuracy with which a numerical value can be read from Graph PFT02C, it is reasonable to conclude that the location of the first zero agrees with the analytical prediction. The occurrence of other zeros on Graph PFT02C at frequencies which are integral multiples of the frequency at the first zero, further supports this conclusion. Examination of the other frequency content graphs in Appendix I yields information in agreement with the conclusion above.

6.2.1.2 Effect of Input Pulse Duration on the Amplitude Ratio and Phase Lag

The effect of input pulse duration on the amplitude ratio and phase lag can be observed by comparing amplitude ratio and phase lag graphs derived from runs 01 through 06 and presented in Section 6.2.1.3.

Three different Fourier transform techniques were used to obtain the graph presented in Section 6.2.1.3 so for each run there are three similar graphs, each derived using a different technique. Because the general effect of input pulse duration was the same regardless of the Fourier transform technique used, no distinction is made between the techniques in the discussion which follows.

As outlined in Section 3.2.2, the frequency at which the first zero of the frequency content curve occurs represents the frequency at which the amplitude ratio and phase lag curves become unreliable. Generally speaking, this is due to division by small numbers in the calculational procedure. The amplitude ratio graphs in Section 6.2.1.3 exhibit this characteristic because large discontinuities occur at frequencies which very closely coincide with the frequencies at the zeros of the frequency content curves for the related input pulses. The amplitude ratio graphs (Section 6.2.1.3) for runs 01, 02 and 03 serve to demonstrate this situation. In Graph PFT01A, there are no discontinuities in the curve up to four radians per second and the related frequency content curve, Graph PFT01C (Appendix I), does not exhibit a zero until approximately 5.85 radians per second. In Graph PFT02A, one discontinuity appears on the curve between two and three radians per second and the related frequency content curve, Graph PFT02C, exhibits zeros at approximately 2.5 and 5.0 radians per second. The discontinuity between two and three radians per second in Graph PFT02A is due to the zero at 2.5 radians per second in the frequency content curve (PFT02C). In the case of Graph PFT03A, three discontinuities appear; between one and two, two and three and three and four radians per second. The zeros

in the related frequency content curve, Graph PFT03C, occur at approximately 1.28, 2.56, 3.84 and 5.12 radians per second. Study of results from other runs yields the same correlation between zeros in the frequency content curve and discontinuities in the amplitude ratio curve.

It should be noted that the discontinuities in the amplitude ratios became less pronounced as the pulse width was increased. This could possibly be due to the reduction in area under the frequency content curve between the zeros. Visual inspection of Graphs PFT02C and PFT04C in Appendix I illustrates this change in area since both graphs have the same scale. For run 02 the input pulse duration was 2.5 seconds while for run 04 it was 10.0 seconds.

When the pulse width became greater than twice the time constant the discontinuities due to the zeros in the frequency content became so numerous they appeared more like random noise. However, from the location of the theoretical high-frequency asymptote on the curves in Graphs PFT04A, PFT05A and PFT06A in Section 6.2.1.3 it appears that the graphical mean of this "noisy signal" is valid representation of the high-frequency portion of the amplitude ratio curve. On the other hand, the amplitude ratio graphs, PFT02A, PFT02A and PFT04A in Section 6.2.1.3, exhibit discontinuities that are distinct and separate and in no way like noise. Here the theoretical high-frequency asymptote intersects the horizontally oriented lines running between the vertical discontinuities at approximately the mid-point. The nature of this intersection would tend to indicate that valid data is being obtained between these discontinuities which implies that valid

data points are being obtained at frequencies between the zeros of the frequency content curve. In support of this point, it is interesting to note that Dreifke (25) was able to obtain valid amplitude ratio results by using only data points calculated at frequencies between the zeros of the frequency content curve. Further, Hougen (40) indicated that Driefke's work clearly demonstrated that valid information could be obtained between these zeros. Clements (12), on the other hand, suggested that information beyond the first zeros was generally unreliable. It, thus, appears from Graphs PFT02A, PFT03A and PFT04A that this work supports Hougen's point of view.

Hougen and Walsh (42) reported that deviations from the true amplitude ratio curve begin to appear when data points are calculated beyond a frequency equal to 80% of the frequency at the first zero of the frequency content curve. This characteristic appears in this work as can be seen from Graph PFT02A in Section 6.2.1.3. The pronounced uncharacteristic rise in the amplitude ratio occurs at a frequency of approximately 2.0 radians per second which is just prior to the first discontinuity. On the related frequency content curve, Graph PFT02C, 80% of the frequency at the first zero is approximately 1.97 radians per second. Therefore, the uncharacteristic rise in the amplitude ratio curve occurs at the frequency predicted by the criterion of Hougen and Walsh (42). Other graphs in Section 6.2.1.3 also exhibit this behaviour.

The phase lag graphs in Section 6.2.1.3 do not exhibit the sharp discontinuities evident in the amplitude ratio graphs. This is because of a breakdown of the subroutine ANGLE as outlined in Section 5.4.2.5. The segmented appearance of some of

the phase lag at low frequencies is due to the fact that data points were calculated at equal linear increments and then plotted on a logarithmic scale by drawing straight lines between these points.

6.2.1.3 Amplitude Ratio and Phase Lag Graphs Derived by Pulse Testing on First Order System

General Comments:

a. Table 2 describes the graphs presented on pages 104-127 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs has been given in Section 6.1.

c. Data presented in these graphs were derived from the time domain pulse data presented in Appendix I.

   Three Fourier transform techniques were used in the derivation:

   1. Fast Fourier transform (FT)
   2. Trapezoidal quadrature (TP)
   3. Filon's quadrature (FN)

d. All amplitude ratio graphs, except Graph PFT02A, are drawn on the same logarithmic-logarithmic scale to facilitate easy comparison.

e. All phase lag graphs, except Graph PFT01B are on the same semi-logarithmic scale.

f. The results obtained from the Fast Fourier transform technique (FT) utilized a time data sample interval equivalent to that on the graphs in Appendix I. The results obtained from either the Filon (FN) or trapezoidal (TP) quadrature utilized an interval that was twice that outlined in Appendix I because of the use of subroutine REDPT (Section 5.4.2.3).

g. The theoretical high-frequency asymptote for the amplitude ratio curve has been drawn on the graphs along with a "+" at the theoretical position of the amplitude ratio curve at the corner frequency. Also, on the phase lag graphs a "+" appears at the theoretical position of the phase lag curve at the corner frequency. The time constant for the first order system was 5.0 seconds in all cases.

h. The graphs on pages 104-127 are presented in the following general order:

   1. amplitude ratio calculated by

      (i) FT: data from pulse tests in order from the narrowest to the widest input pulse.

      (ii) TP: as under (i).

      (iii) FN: as under (i).

   2. phase lag calculated by: (i), (ii) and (iii) as under 1.

6.2.2 <u>A comparison of Fourier Transform Techniques</u>

In order to obtain frequency response data from time domain pulse data, it is necessary to obtain a numerical approximation to the Fourier transform of the input and the output pulse. In this work, three techniques were used to evaluate the Fourier transform:

a. Fast Fourier Transform (FFT)

b. Trapezoidal quadrature (TP)

c. Filon's quadrature (FN)

TABLE 2

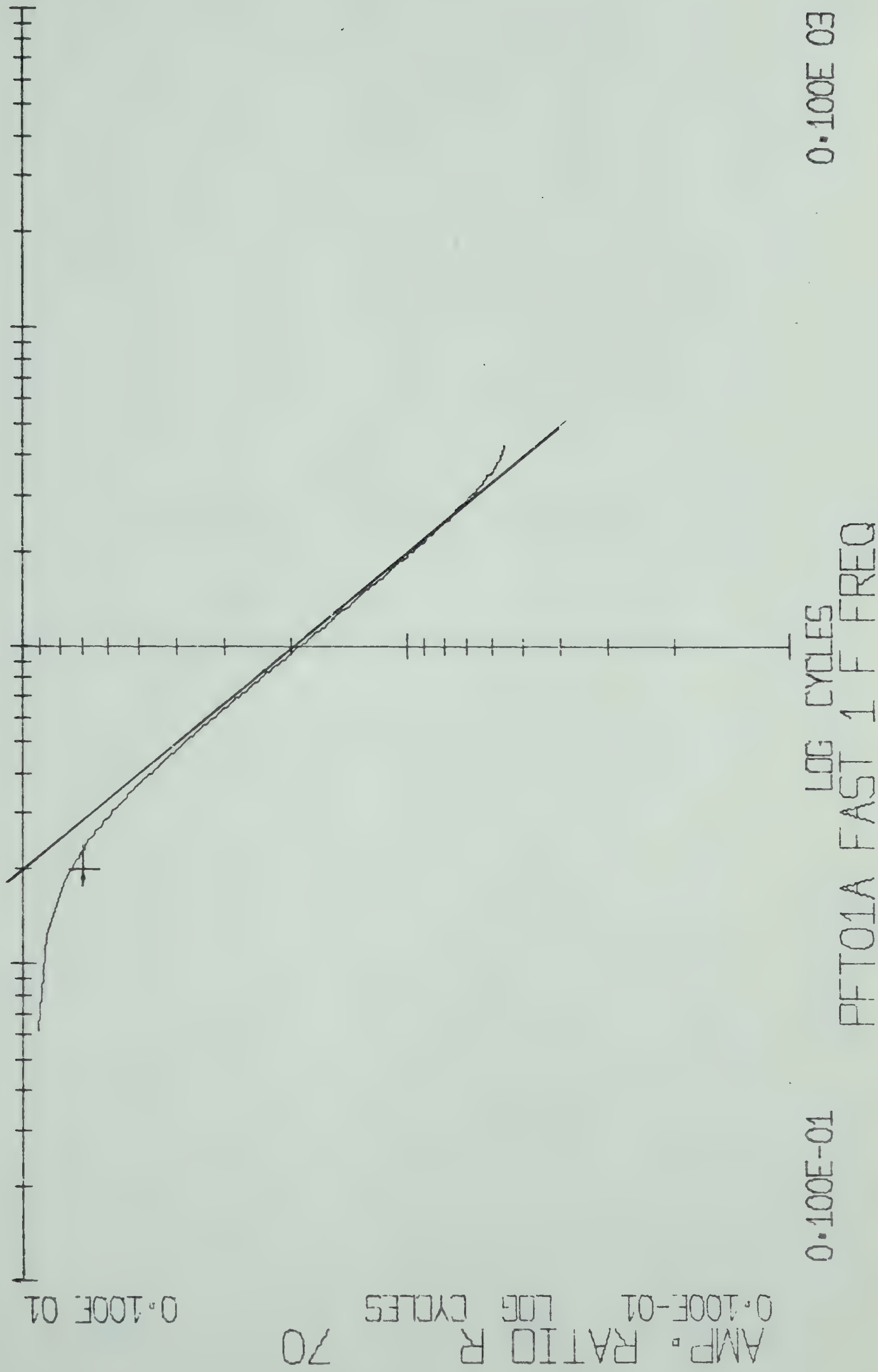
LIST OF FIRST ORDER SYSTEM FREQUENCY RESPONSE GRAPHS

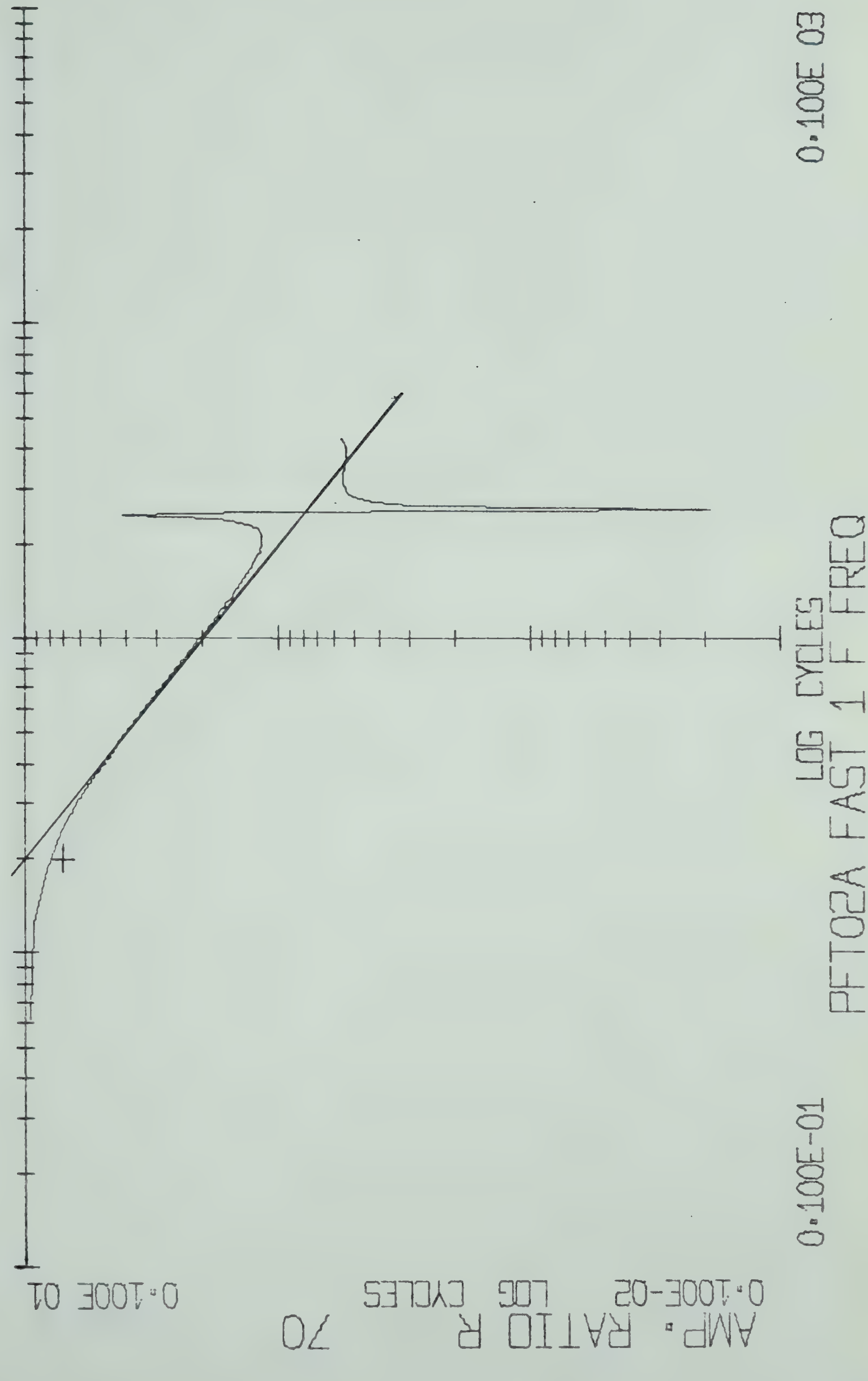
GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT01A	104	1.0	FT	Amplitude ratio
PFT02A	105	2.5	FT	Amplitude ratio
PFT03A	106	5.0	FT	Amplitude ratio
PFT04A	107	10.0	FT	Amplitude ratio
PFT05A	108	25.0	FT	Amplitude ratio
PFT06A	109	75.0	FT	Amplitude ratio
PTP01A	110	1.0	TP	Amplitude ratio
PTP02A	111	2.5	TP	Amplitude ratio
PTP03A	112	5.0	TP	Amplitude ratio
PTP04A	113	10.0	TP	Amplitude ratio
PTP05A	114	25.0	TP	Amplitude ratio
PFN01A	115	1.0	FN	Amplitude ratio
PFN02A	116	2.5	FN	Amplitude ratio
PFN03A	117	5.0	FN	Amplitude ratio
PFN04A	118	10.0	FN	Amplitude ratio
PFN05A	119	25.0	FN	Amplitude ratio
PFN06A	120	75.0	FN	Amplitude ratio

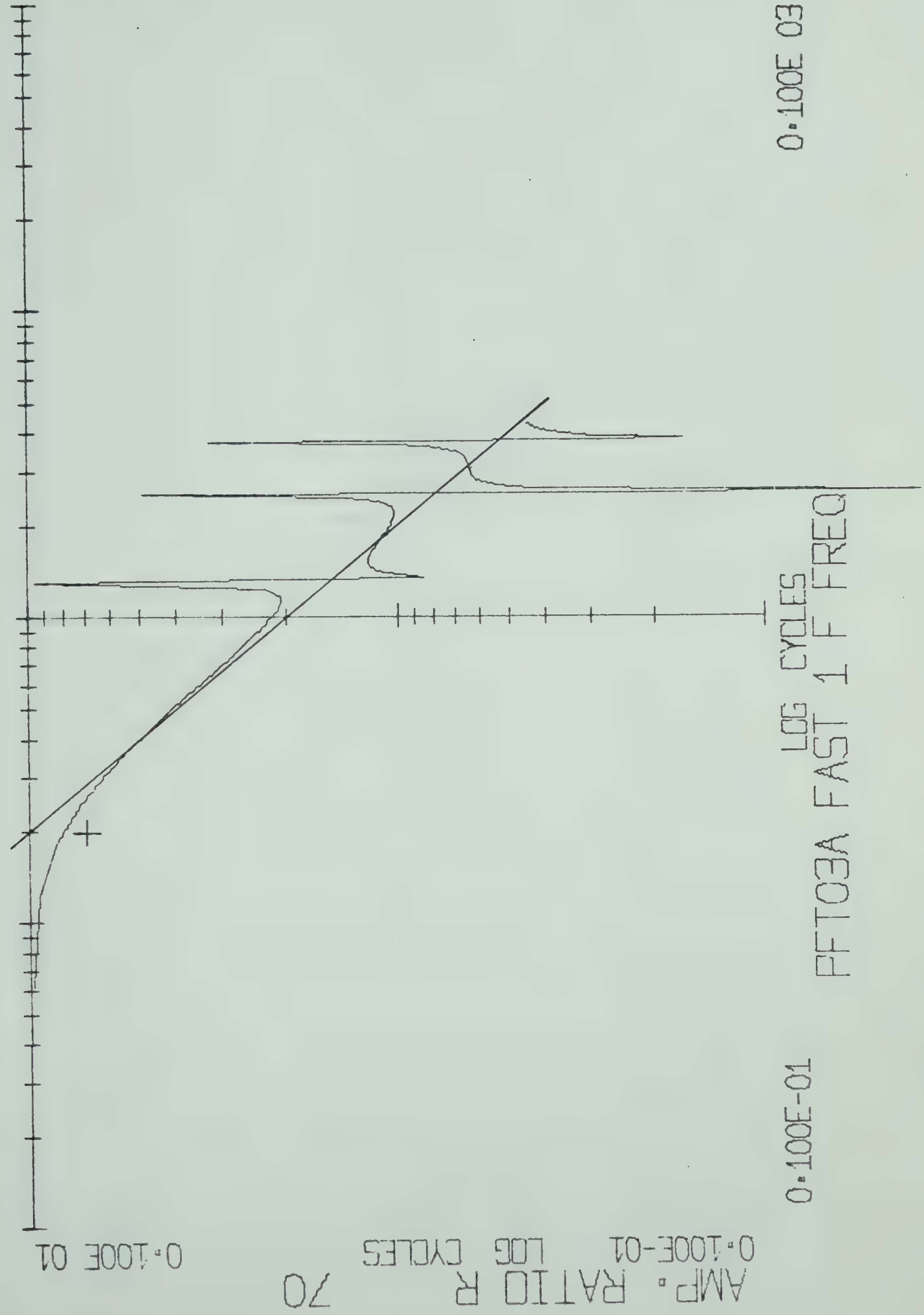
TABLE 2 (CONT'D)

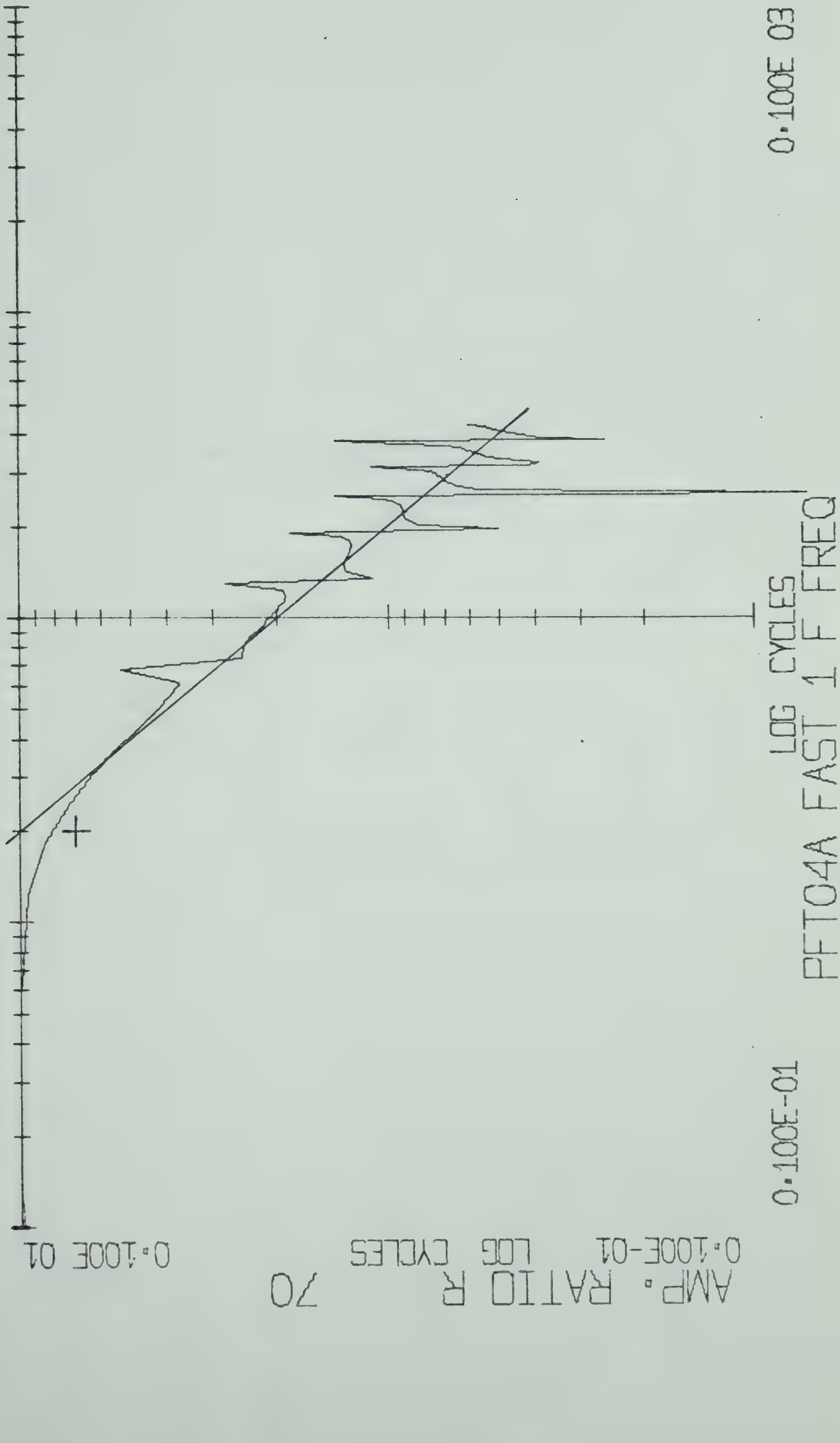
LIST OF FIRST ORDER SYSTEM FREQUENCY RESPONSE GRAPHS

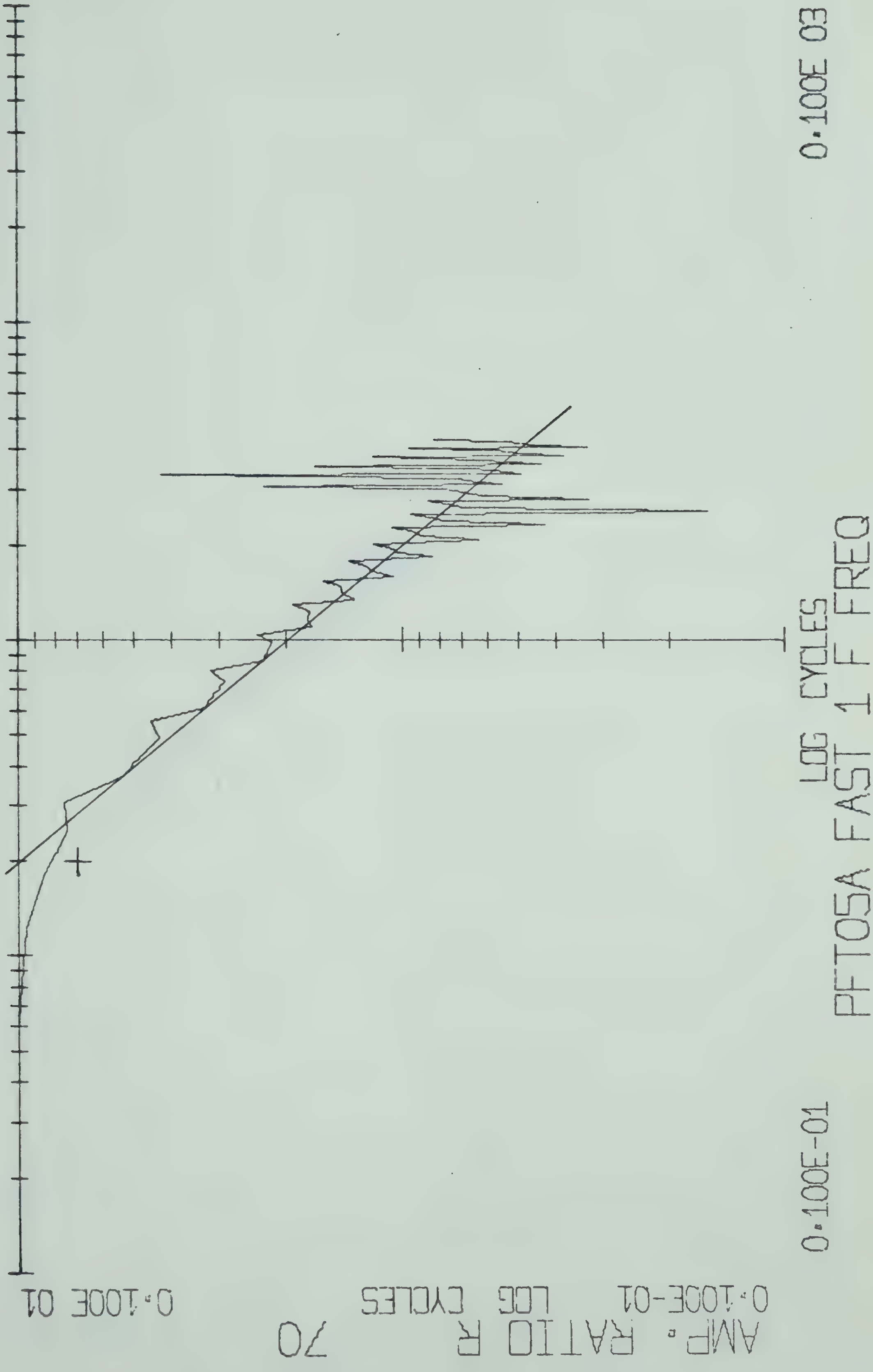
GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT01B	121	1.0	FT	Phase lag
PFT03B	122	5.0	FT	Phase lag
PFT06B	123	75.0	FT	Phase lag
PTP01B	124	1.0	TP	Phase lag
PTP03B	125	5.0	TP	Phase lag
PFN01B	126	1.0	FN	Phase lag
PFN03B	127	5.0	FN	Phase lag

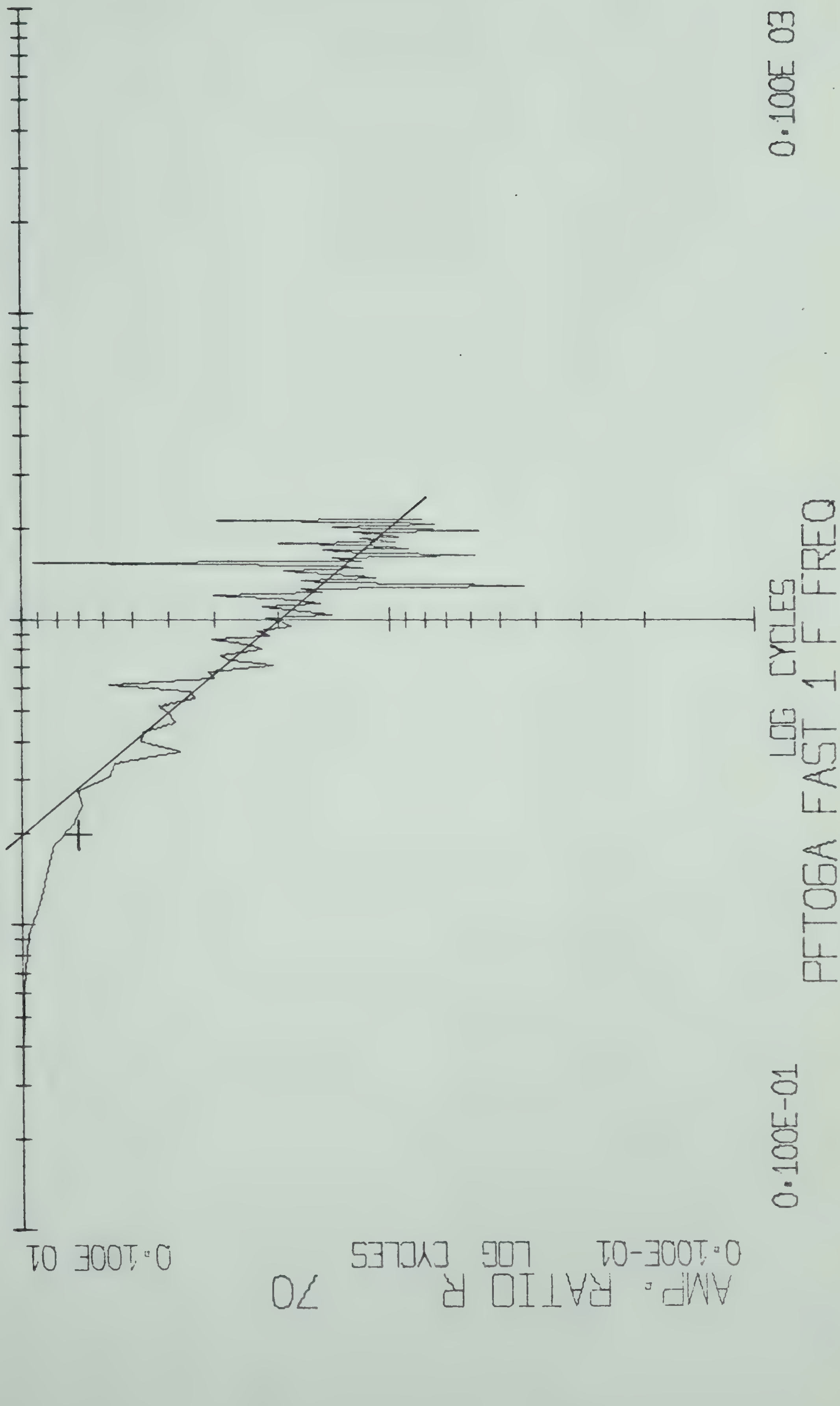

0.100E 03
LOG CYCLES
PFT01A FAST 1 F FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70

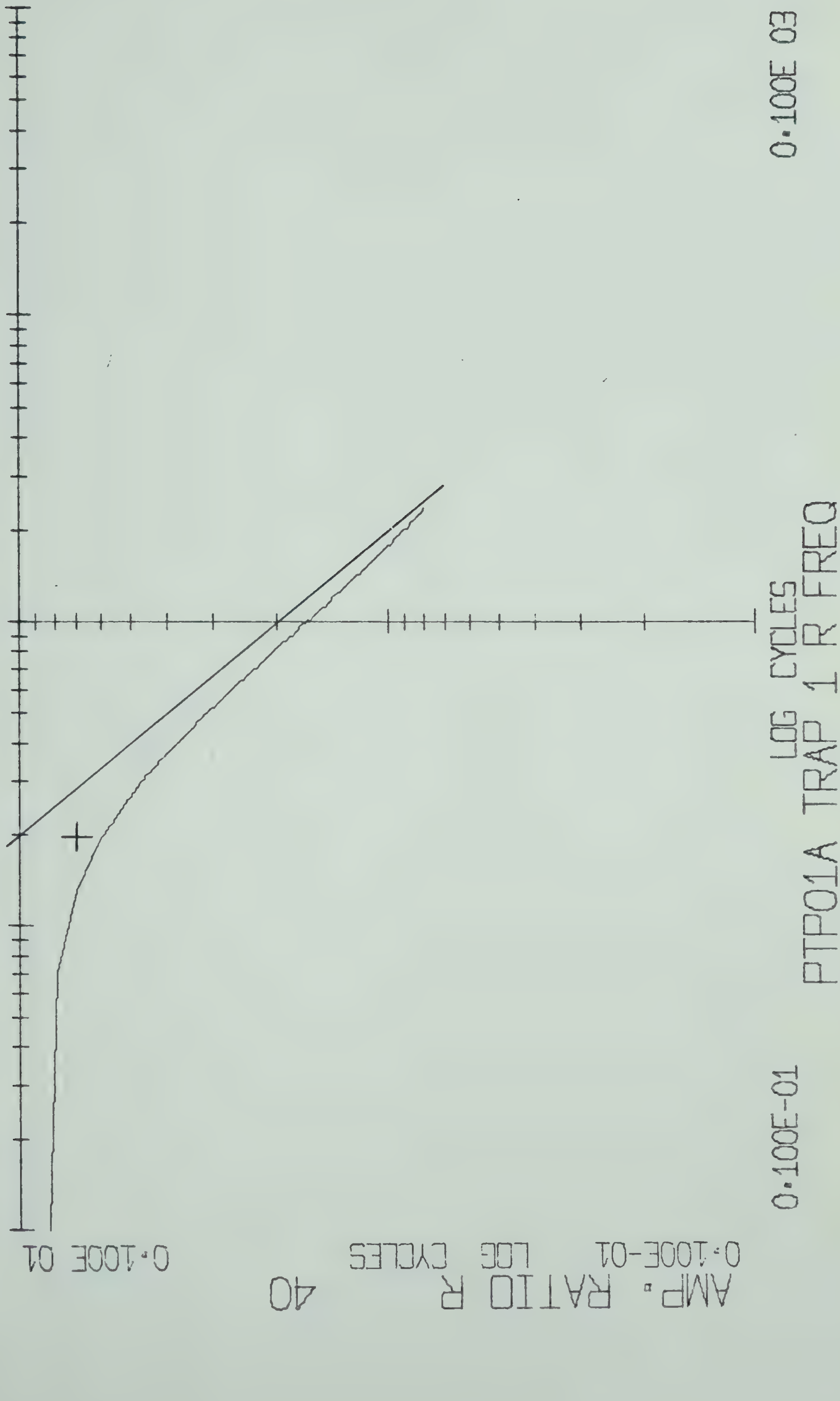

0.100E 03
0.100E-01
LOG CYCLES
PFT02A FAST 1 F FREQ
0.100E 01
0.100E-02
LOG CYCLES
AMP. RATIO R
70

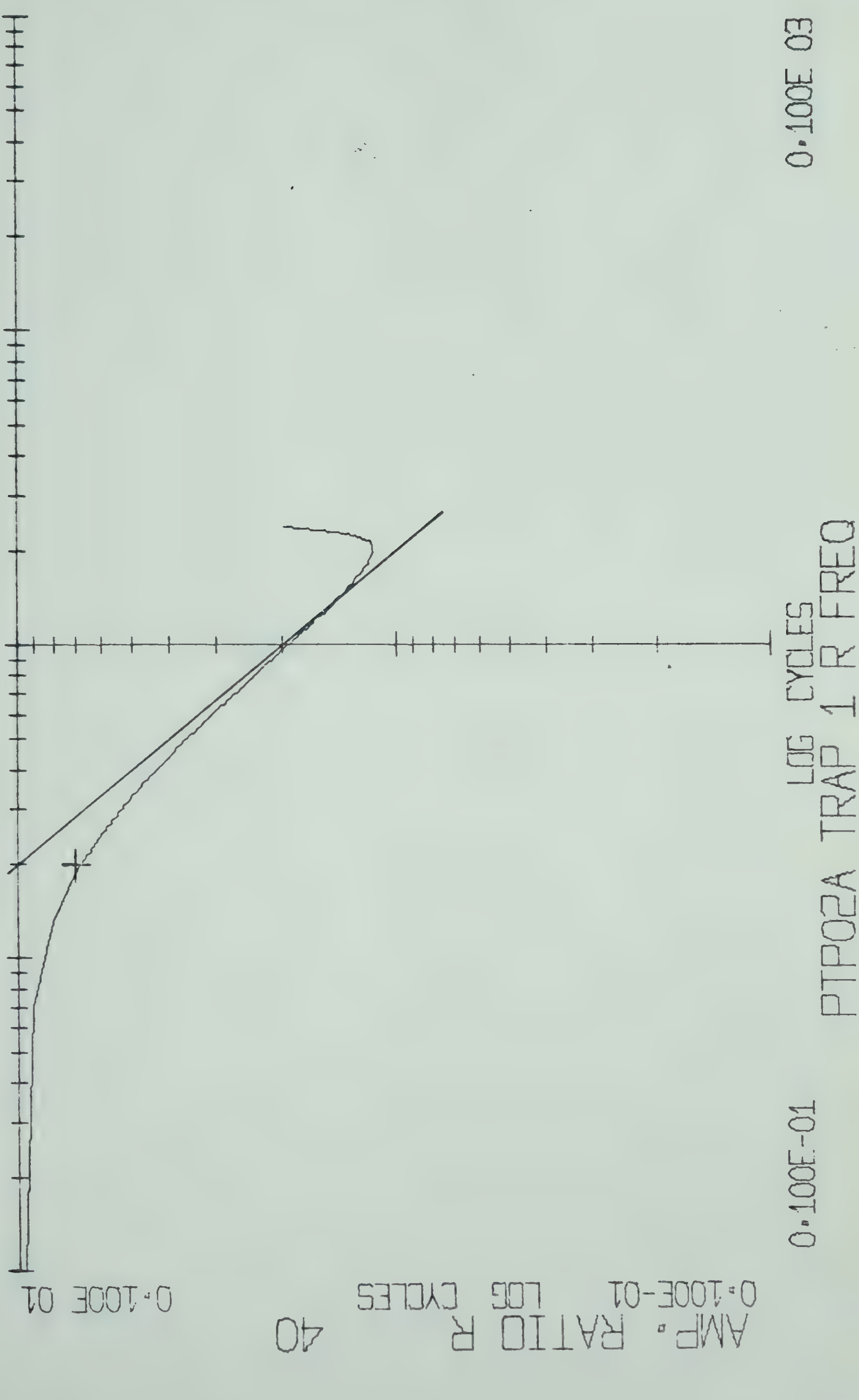

0.100E-01
0.100E 03
LOG CYCLES
PFT03A FAST 1 F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70

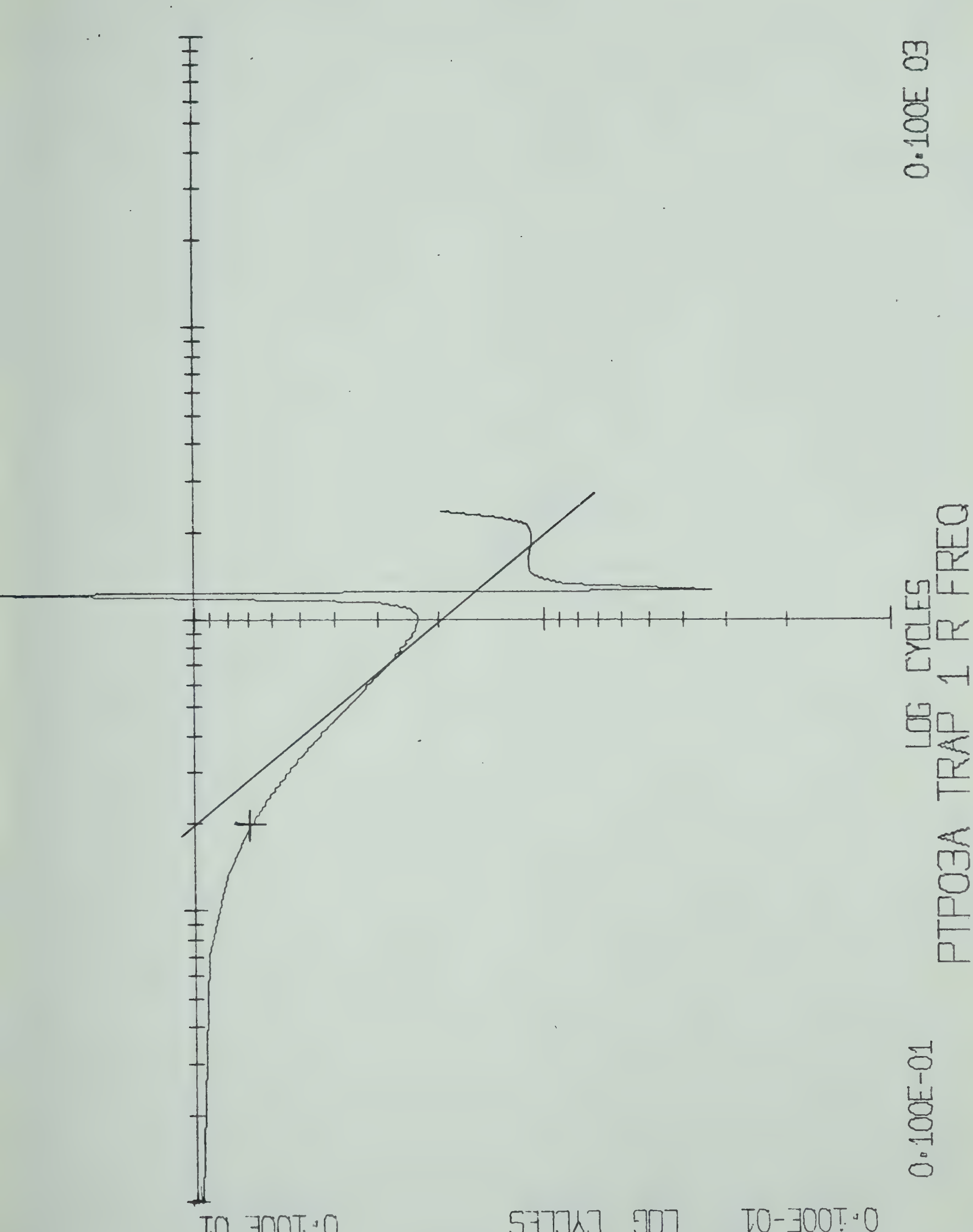

0.100E 03
0.100E-01
LOG CYCLES
PFT04A FAST 1 F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70

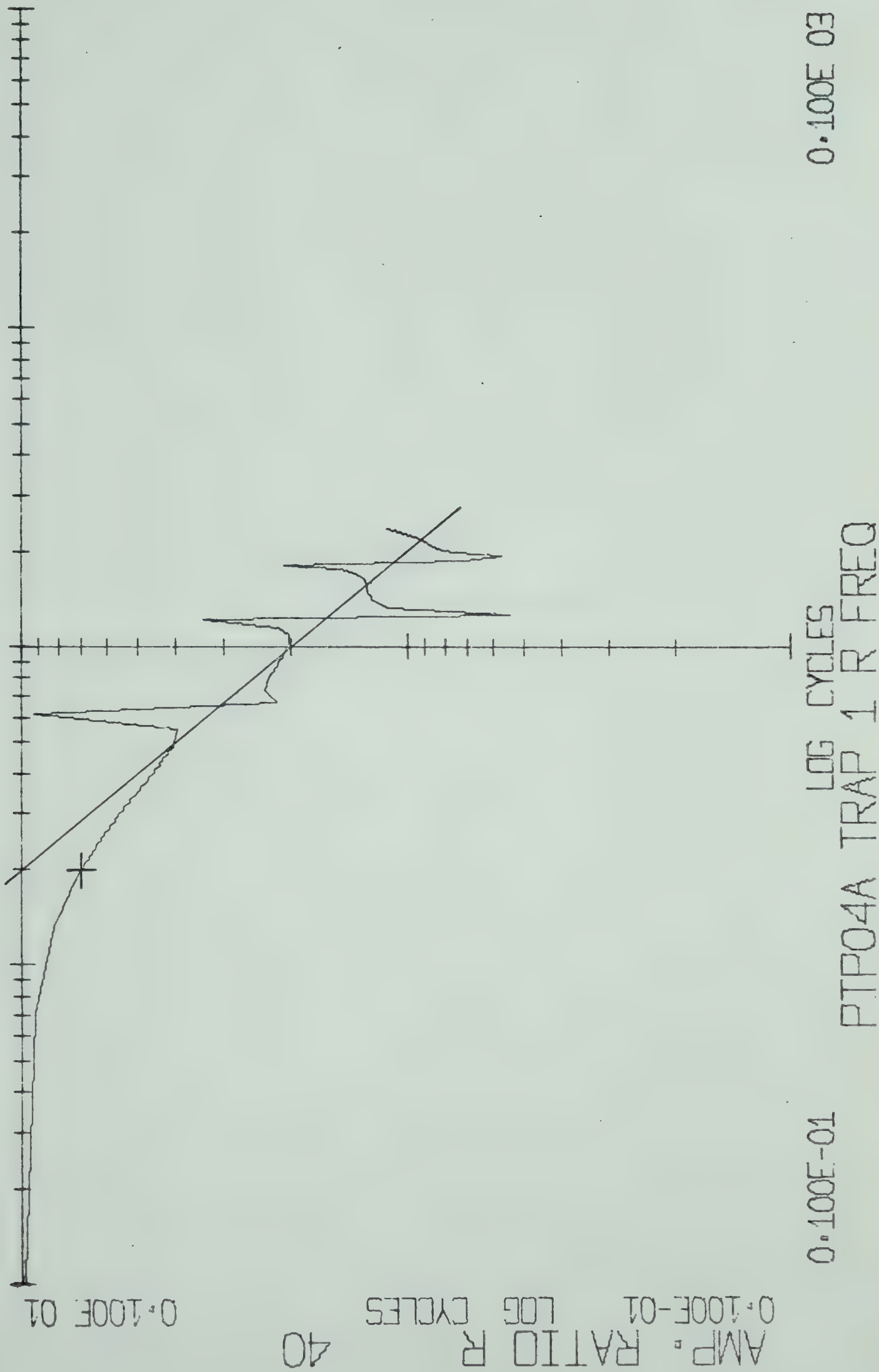

0.100E-01
0.100E 03
LOG CYCLES
PFTO5A FAST 1 F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70

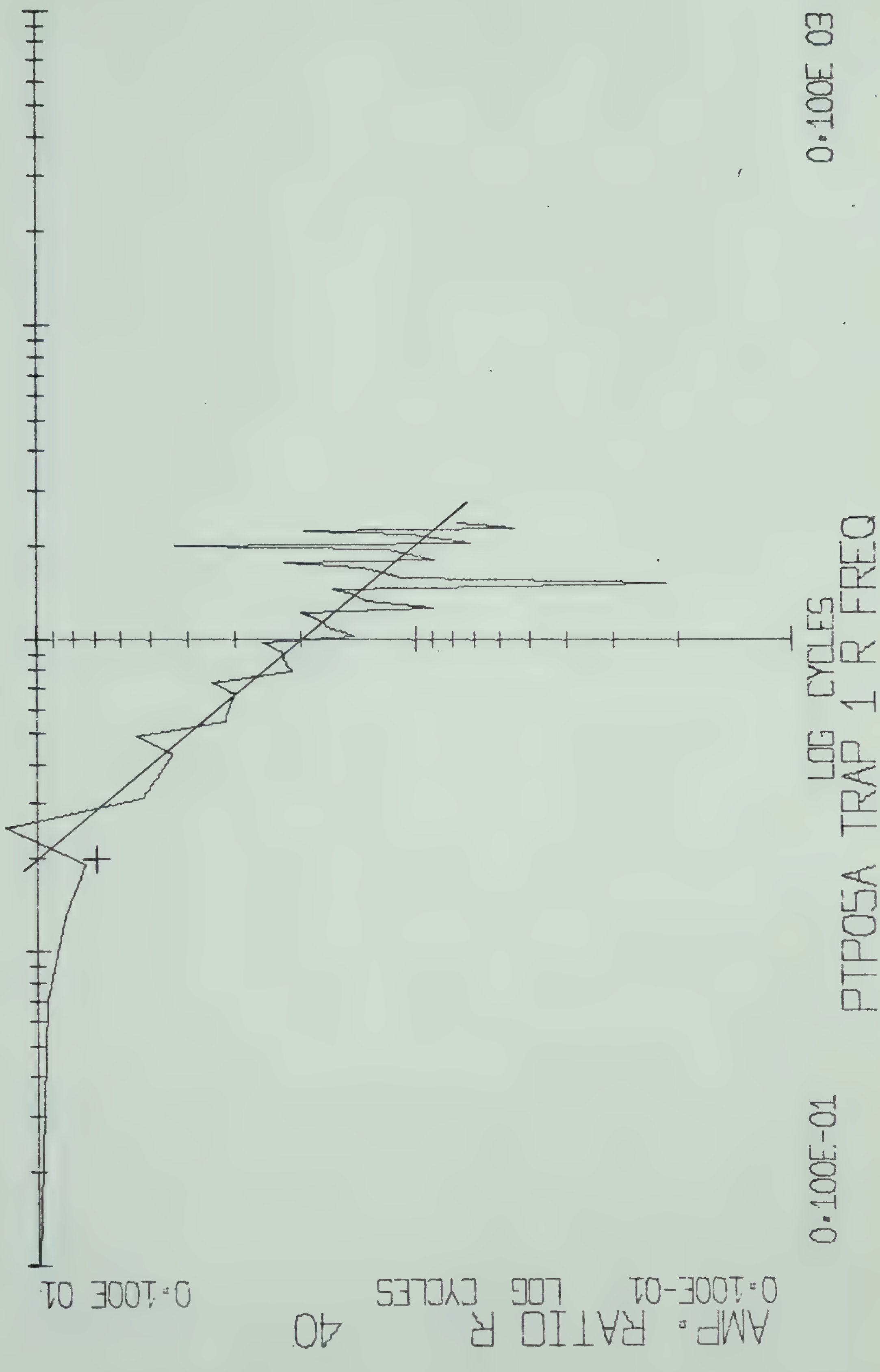

0.100E 03
LOG CYCLES
PFTOGA FAST 1 F FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70

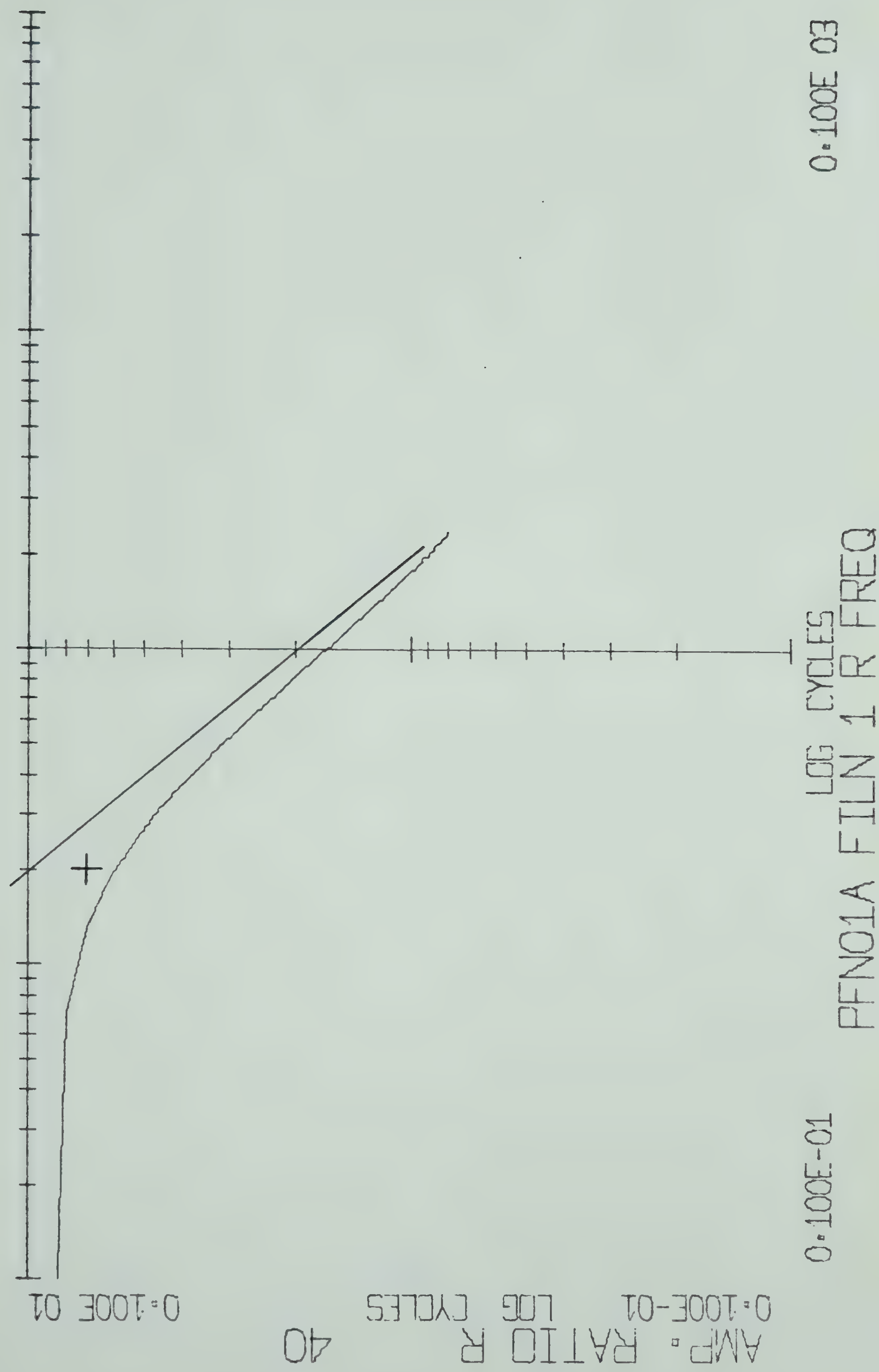

0.100E 03
LOG CYCLES
PTP01A TRAP 1 R FREQ
0.100E-01
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R
40

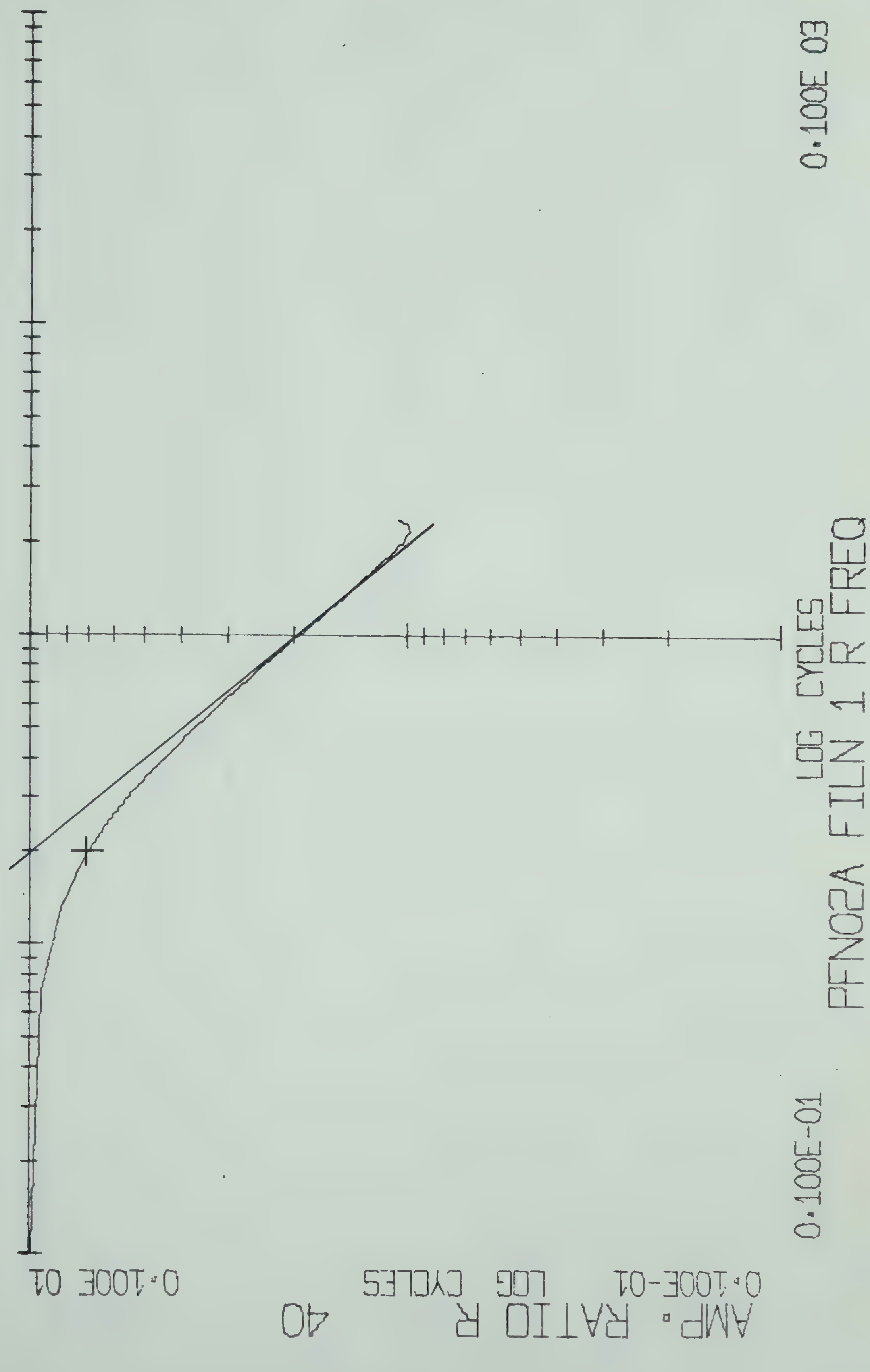

0.100E 03
LOG CYCLES
PTP02A TRAP 1 R FREQ
0.100E-01
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

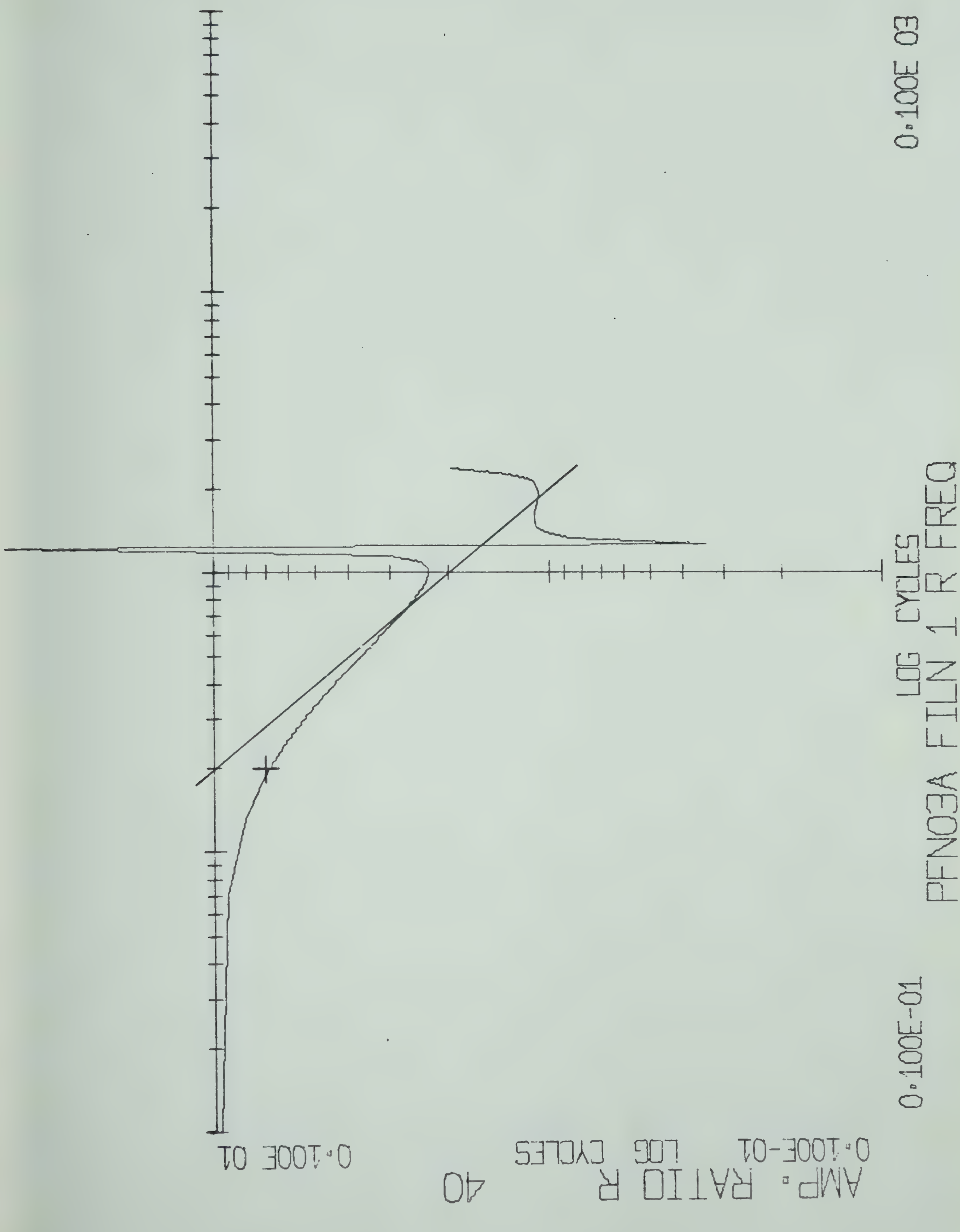

0.100E-01
0.100E 03
LOG CYCLES
PTPO3A TRAP 1 R FREQ
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

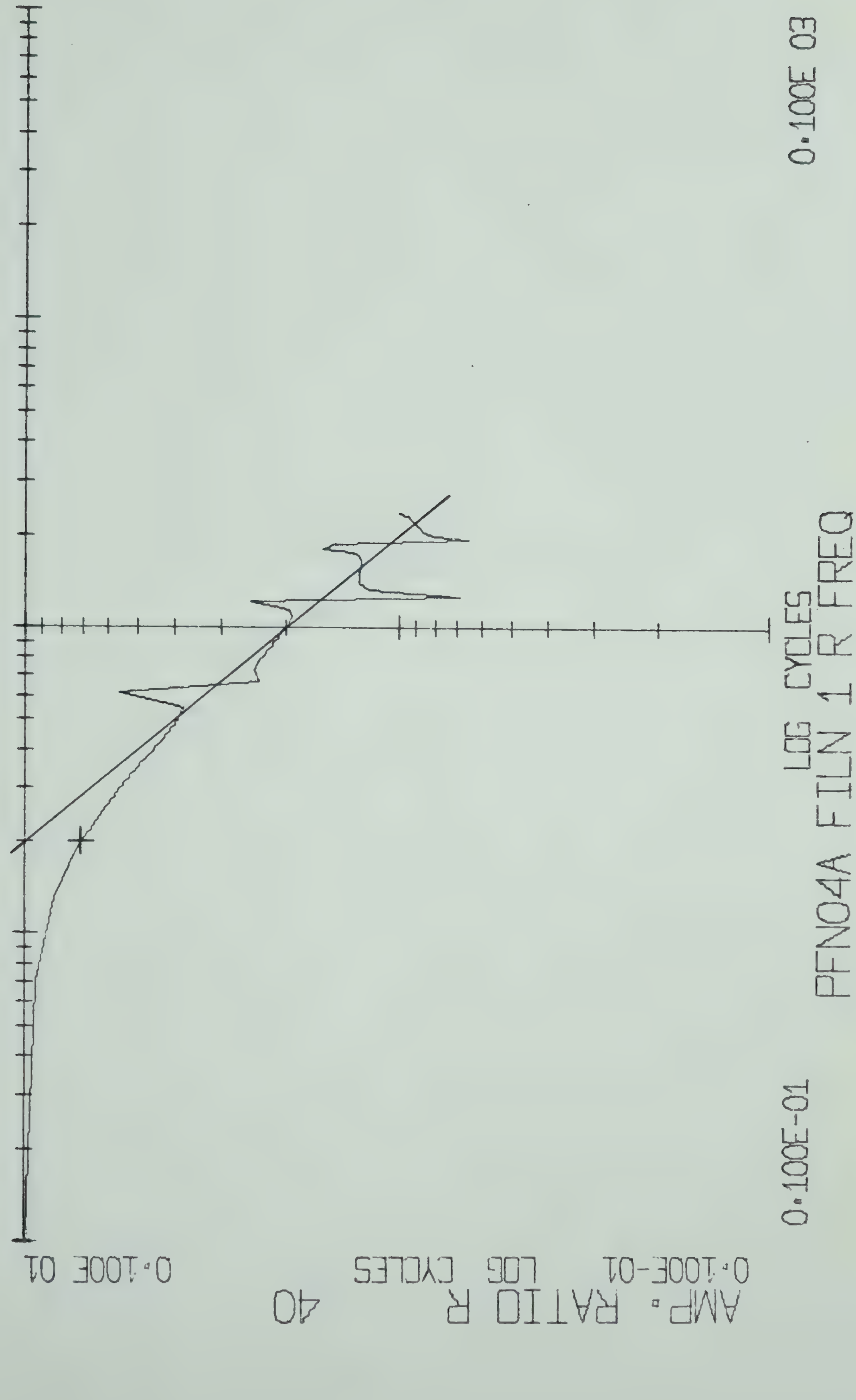

0.100E-01
0.100E 03
LOG CYCLES
PTPO4A TRAP 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R 40

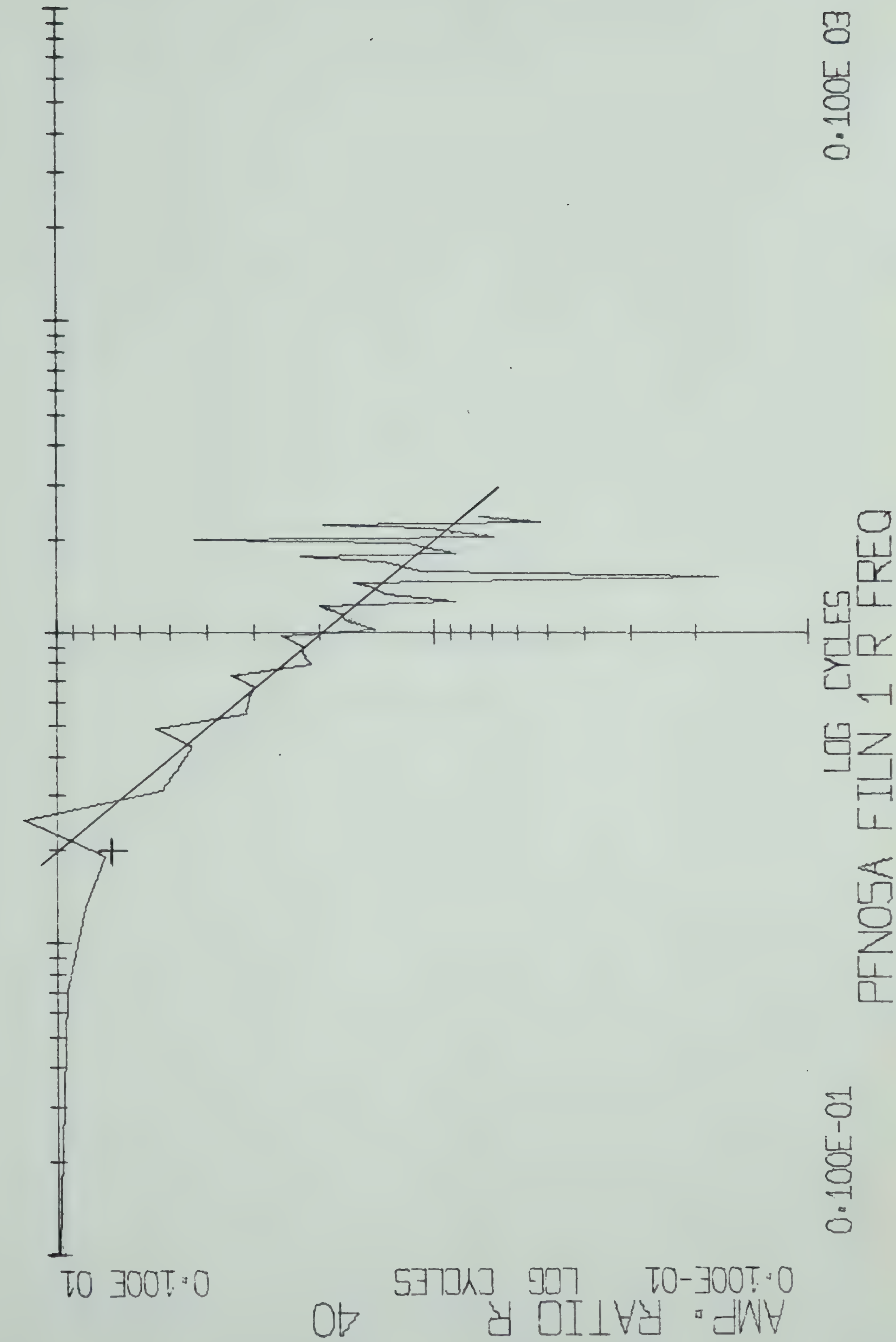

0.100E 03
LOG CYCLES
PTPO5A TRAP 1 R FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

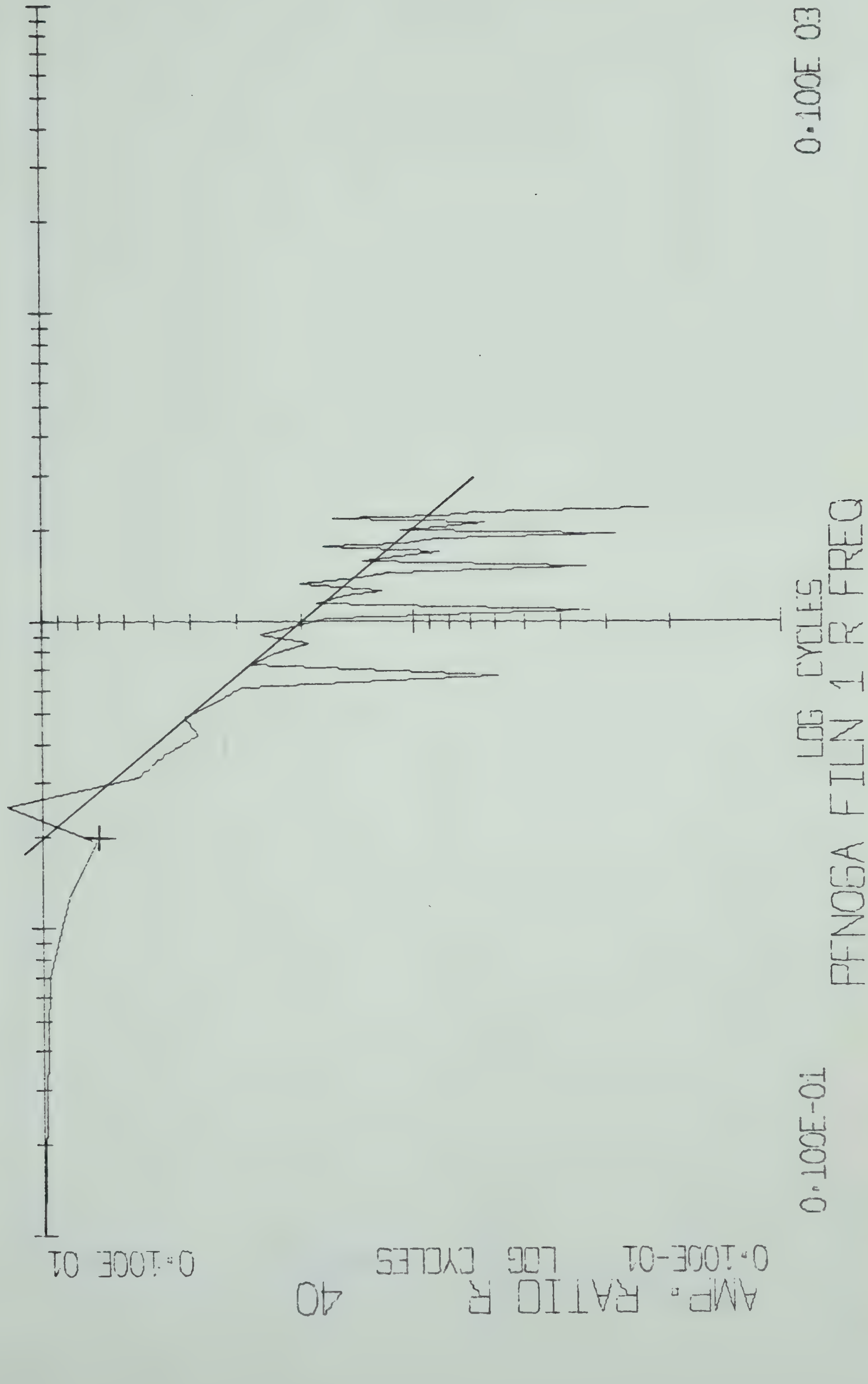

0.100E 03
0.100E-01
LOG CYCLES
PFNO1A FILN 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

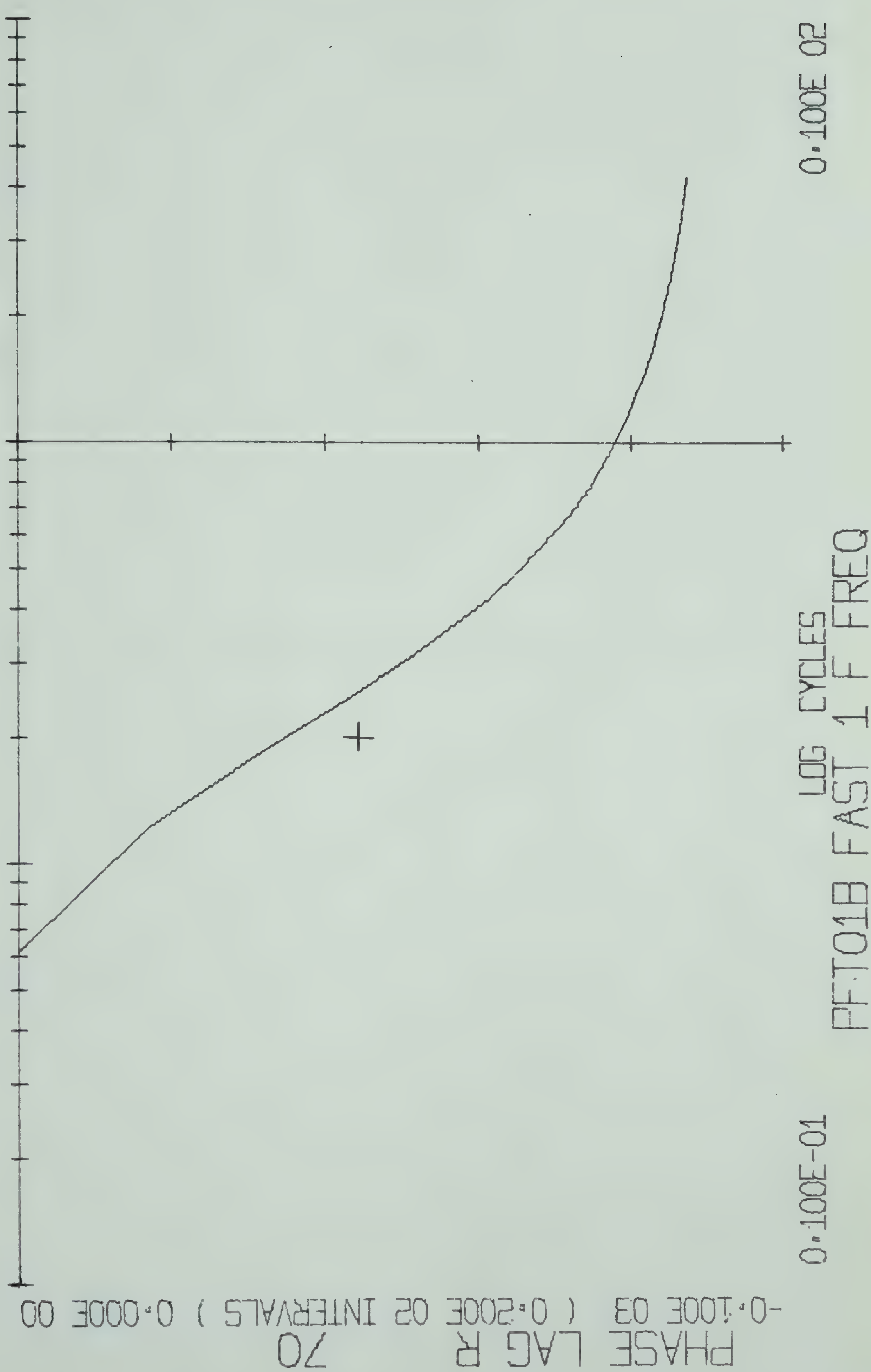

0.100E-01
0.100E 03
LOG CYCLES
PFNO2A FILN 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

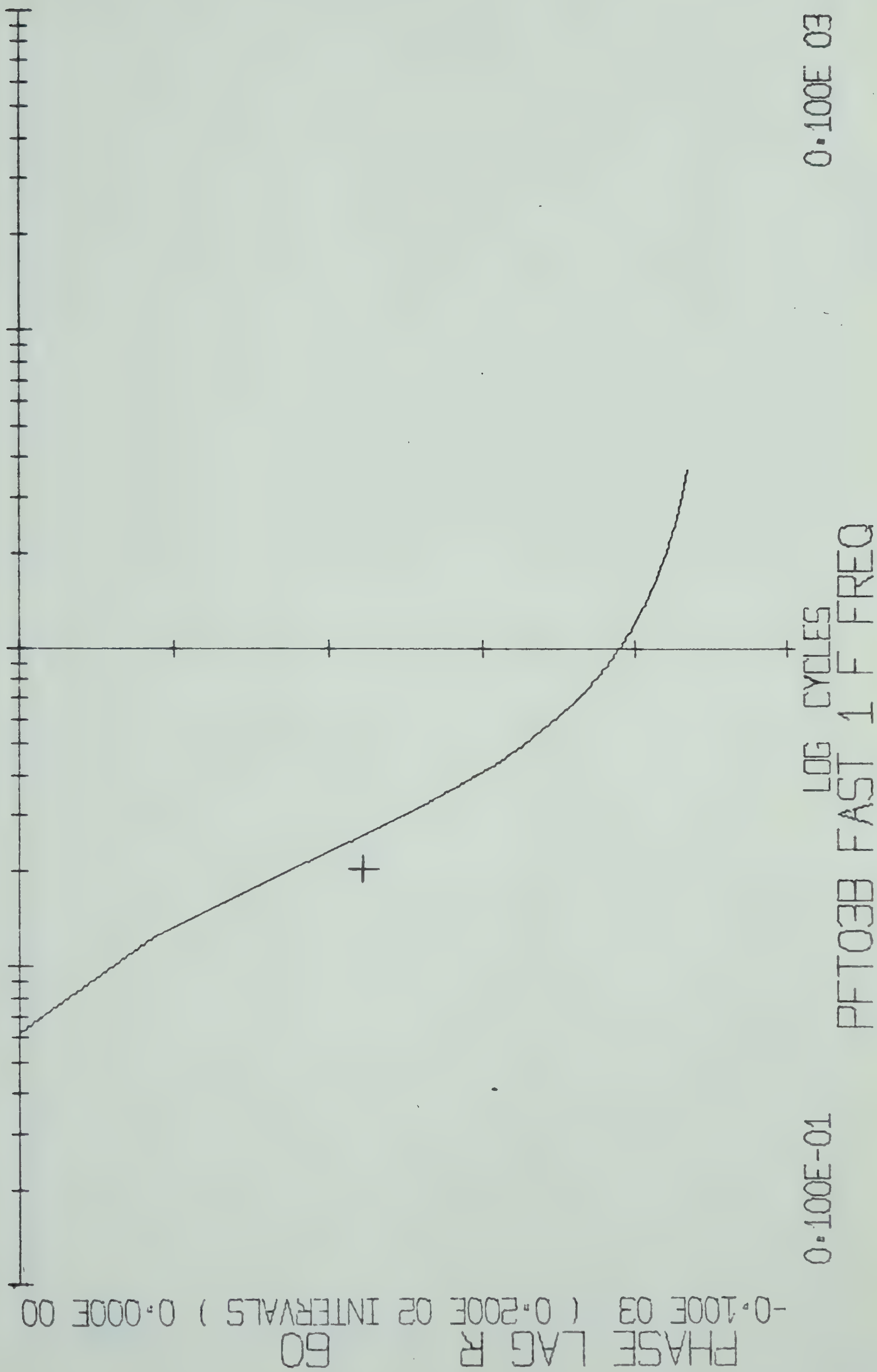

0.100E 03
LOG CYCLES
PFN03A FILN 1 R FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

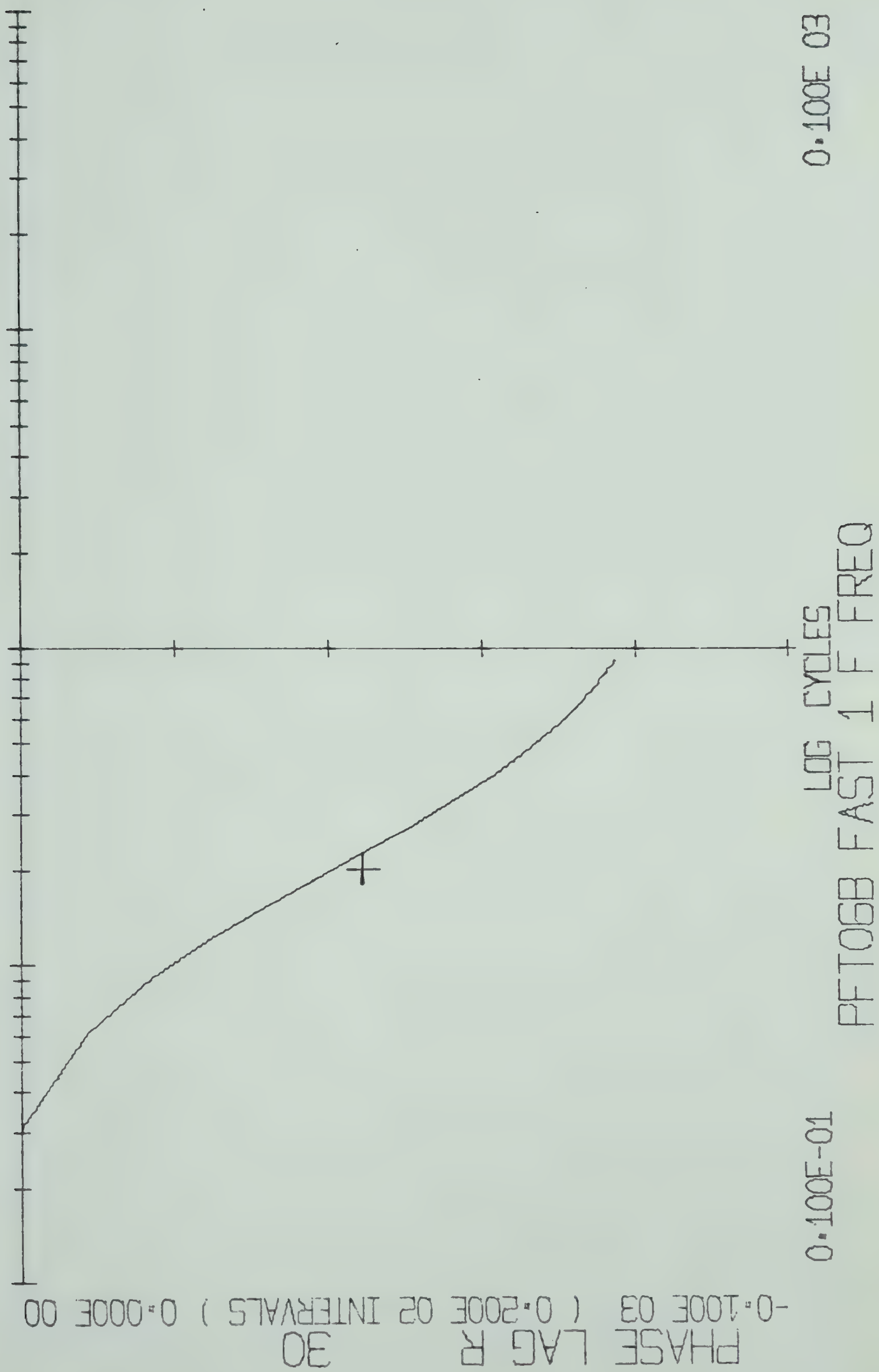

0.100E 03
0.100E-01
LOG CYCLES
PFNO4A FILN 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

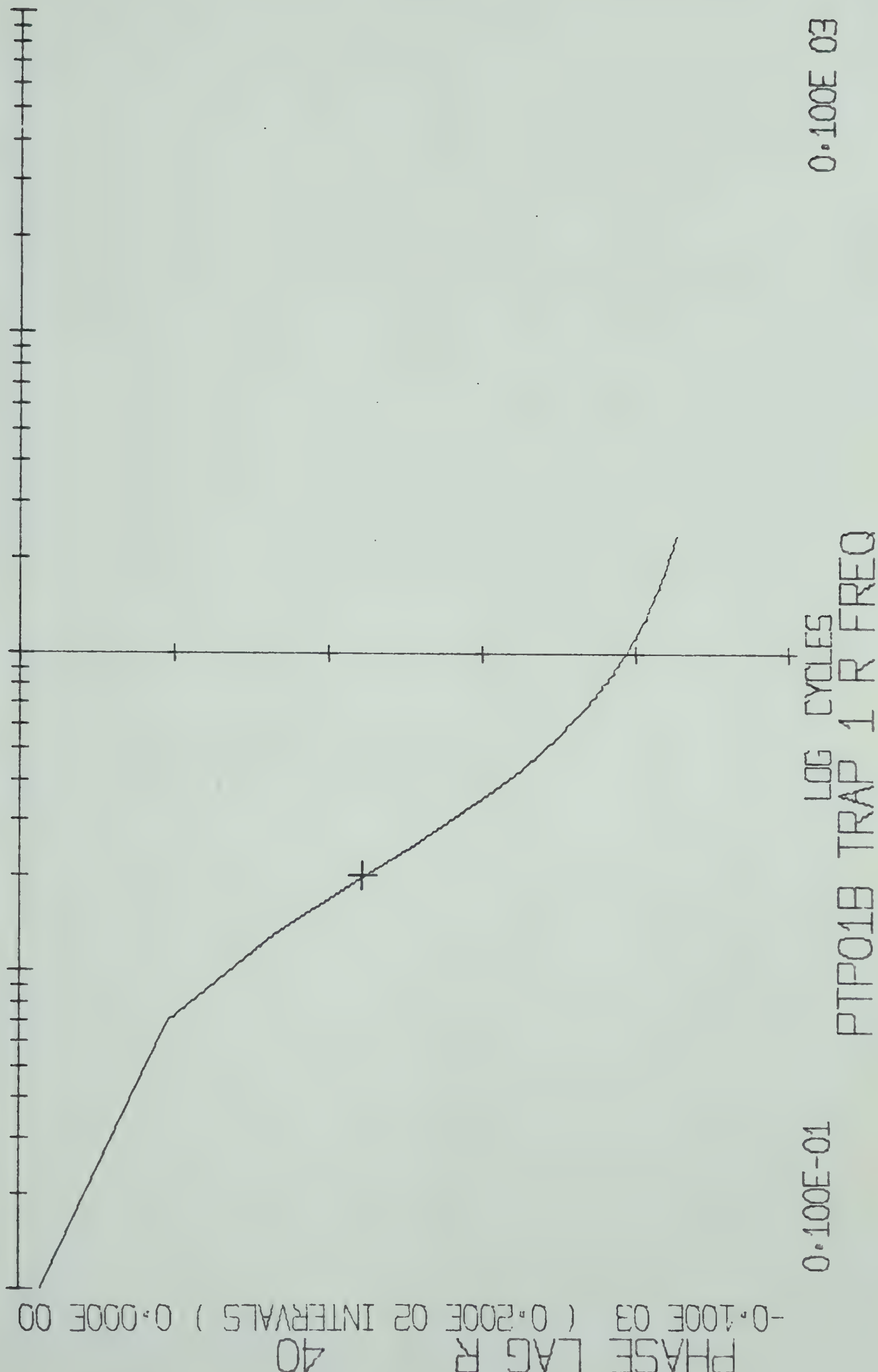

0.100E 03
LOG CYCLES
PENOSA FILN 1 R FREQ
0.100E-01
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

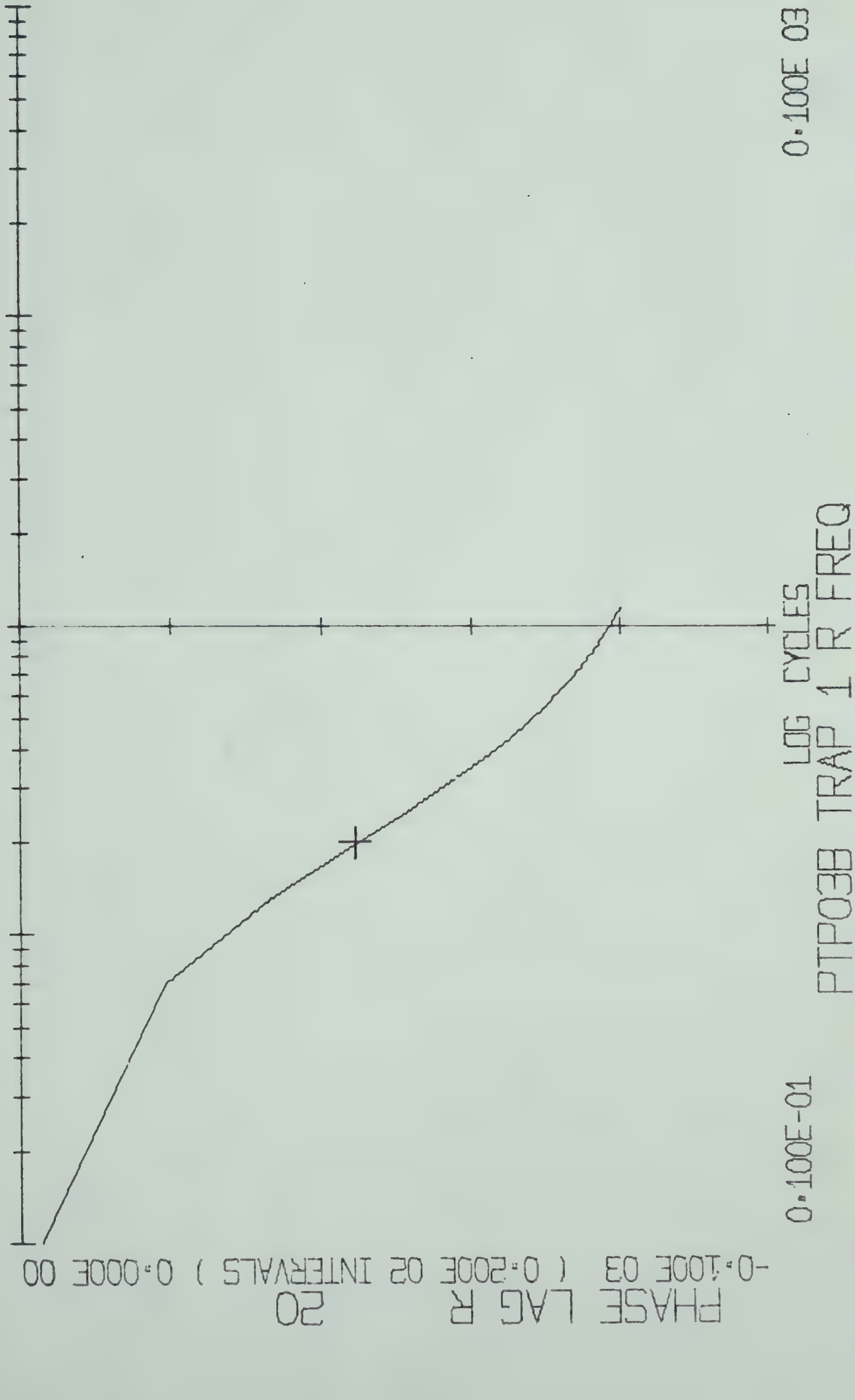

0.100E 03
0.100E-01
LOG CYCLES
PFNOGA FILN 1 R FREQ
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

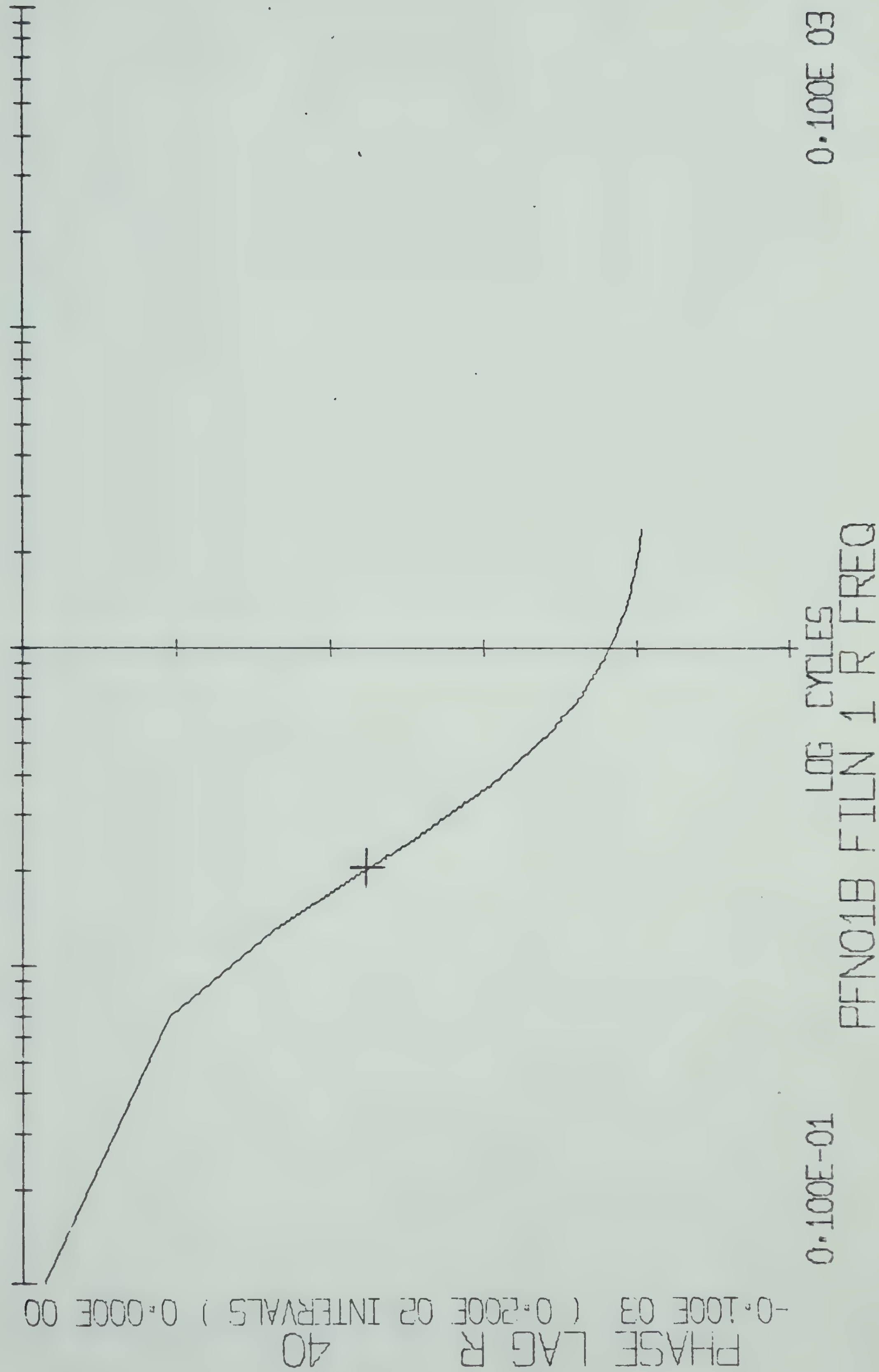

0.100E 02
LOG CYCLES
PFT01B FAST 1 F FREQ
0.100E-01
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R 70

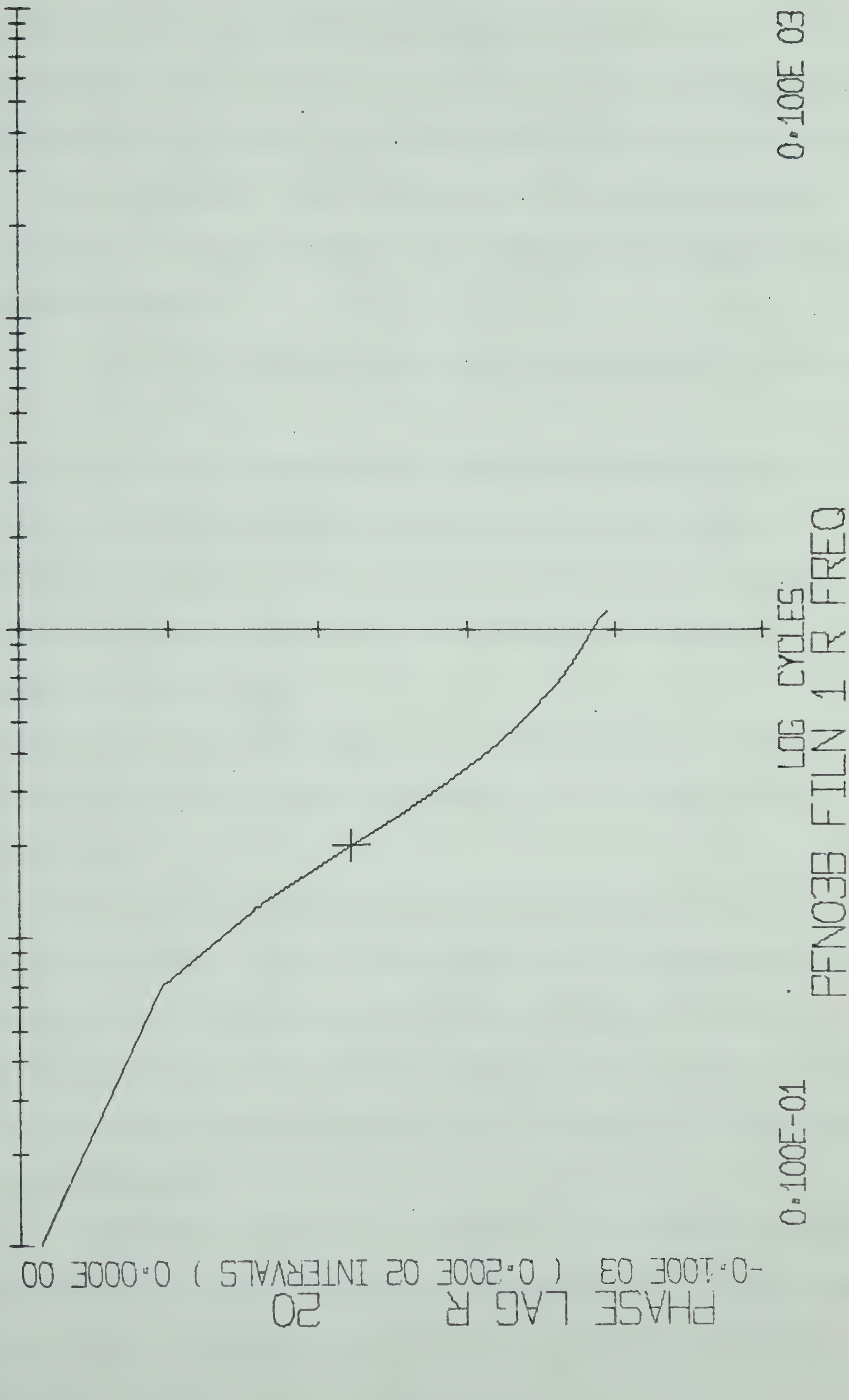

0.100E-01
0.100E 03
LOG CYCLES
PFT03B FAST 1 F FREQ
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R 60


0.100E 03
LOG CYCLES
PFTOGB FAST 1 F FREQ
0.100E-01
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R 30


PTP01B TRAP 1 R FREQ
LOG CYCLES
0.100E-01
0.100E 03
PHASE LAG R
40
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00


0.100E 03
LOG CYCLES
PTP03B TRAP 1 R FREQ
0.100E-01
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R
20


0.100E 03
0.100E-01
LOG CYCLES
PFNO1B FILN 1 R FREQ
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R 40


0.100E 03
0.100E-01
LOG CYCLES
PFNO3B FILN 1 R FREQ
-0.100E 03 (0.200E 02 INTERVALS) 0.000E 00
PHASE LAG R
20

The background of these three techniques has been discussed in Section 3.2.4 and the computer programs for their implementation has been discussed in Section 5.4.2.3. Generally speaking, all three techniques provided valid results for the cases considered in this work and consequently, the advantages and disadvantages of each technique appear to be more related to the implementation aspects than to the results yielded.

The FFT is distinct from either of the other two techniques because:

a. It has a one to one relationship between the number of time data points used and the number of frequency points calculated.
b. The minimum frequency value and the increment between frequency points is fixed by the number of time data points and the sampling interval used.
c. The number of time data points used must be a power of two and the sampling interval must be constant over the range of the time data.

Because of the above requirements, careful design of each pulse test is necessary when the FFT is used. If the number of time data points and the sampling interval are not properly chosen, frequency information will not be obtained over the range of interest. These factors combined to restrict the span of frequency data that resulted from the pulse tests of the heat exchanger.

During the preliminary evaluation of the FFT it was found that fewer than 512 time points yielded a frequency band of insufficient width and, therefore, in most cases reported herein, 1024 time points

were used. It should be noted, however, that no attempt was made to increase the time sample interval to the largest value which would adequately characterize the input pulse or the process tested whichever was limiting. An increase in sample interval would have compensated to some degree for the effect of a reduction in the number of time data points (Section 3.2.4). However, even with the maximum sample interval, it is the author's opinion that the number of time points desirable would be in the order of 256 or more. For this reason, the FFT does not appear to be a desirable technique to use when data has been obtained from an analog record and must be punched on computer cards for processing. In an application where the data is acquired directly from the process, as it was in this work, the FFT becomes more advantageous because the computing time involved appears in the order of four to five times less than that required by each of the other techniques. This time comparison is based upon the time required to obtain data suitable for plotting the frequency response. Usually equivalent results were obtained in the work using 1024 time points for the FFT and 360 time points for each of the other techniques.

In summary, the FFT appears to be undesirable to use unless an on-line data acquisition system is available. With on-line data acquisition, the only disadvantage is the need for careful experimental design which may lead to undesirable compromises for some applications.

The program BFKTY described in Section 5.4.2.5 contained an error in the equation which calculated the frequency content curve. The equation should have read:

$$F.C. = \left[A^2 + B^2 \right] \cdot DT/AREA$$ but was programmed

$$F.C. = \left[A^2 + B^2 \right] /N/DT/AREA.$$ By its nature, the error affected the ordinate scaling on some of the frequency content curves but did not affect the shape of the curve. The frequency content is normalized in all cases, so that the value of the ordinate at zero frequency should be very close to one. When the scaling is labelled otherwise, it should be disregarded and a value of 1.00 should be considered at the point where the curve meets the ordinate axis.

The trapezoidal (TP) and Filon (FN) techniques provided what appeared to be identical frequency domain curves for any given set of pulse data. The degree of this similarity can be observed by comparing the associated graphs in Section 6.2.1.3. Because this similarity was common to the results from all runs considered in this work, the results of the TP or FN technique, presented in Sections other than 6.2.1.3, were selected arbitrarily to represent the results from both techniques.

The computing time was approximately the same for both techniques so the results in this work did not demonstrate any advantages for one technique over the other. However, Clements (12) compared these two techniques in greater detail than was considered here and reported a slight advantage for the FN technique. The principal reasons for this advantage were stated to be:

a. TP is sensitive to higher frequencies; FN is not (Section 3.2.4).

b. FN is less sensitive to pulse data with limited accuracy or random errors.

c. FN is less sensitive to the use of a large sampling interval on the tail of the pulse.

In the light of this work and that of Clements, it appears reasonable to conclude that the FN technique has a marginal advantage over the TP technique.

Both the FN and TP techniques are completely flexible with respect to selection of the frequency points to be calculated and the sample interval used for the pulse data. In most applications reported in the literature, two sample intervals were used to characterize the pulse curve with a narrow interval used on the steep rise of the curve and a longer interval used on the slowly decaying tail. The two intervals made it possible to reduce the number of time samples used which saved on both computing time and data preparation time when the data was entered from cards. It can be seen that the above-mentioned flexibility makes the TP and FN techniques more advantageous than the FFT when the pulse data has to be entered using hand punched cards. Clements (12) indicated that between fifty and one hundred data points per pulse curve are all that are required when the TP or FN technique is applied using a two sample interval program. In this work a single sample interval was used because the data were acquired directly from the process and computing time was not at a premium.

The results obtained in this work using the three Fourier transforms are presented in Section 6.2.1.3. They exhibit some minor differences but no definite conclusions can be drawn on the basis of this limited investigation. From a study of the high-frequency asymptotes

it appears that the slope of the amplitude ratio curve is good for all the graphs when consideration is given to the effect of the zeros as discussed in Section 6.2.1.2. However, examination of the FT results shows that the amplitude ratio curve is generally high and the phase lag curve is generally to the right of the true curve at the corner frequency. In the case of the TP and FN techniques, the agreement between the calculated curve and the true curves at the corner frequency is good for both the amplitude ratio and the phase lag except for the results from runs 01 and 02 No explanation can be given for the consistent shift that is demonstrated in the results of the FFT and until future work determines the cause, it can only be stated that the FFT does not yield results as accurate as those obtained from TP and FN.

It was observed in this work that the size of the sample interval used for the pulse data slightly affected the position of the curve on the amplitude ratio graph regardless of the transform technique used. No time was available to investigate this effect. It is interesting to note that the amplitude ratio curves do not always approach unity as the frequency approaches zero for the graphs in Section 6.2.1.3. In fact, the narrower the input pulse the farther the low frequency component of the curve lies below the unity axis. This would imply that the sample interval used (constant for all runs except 06) was too long to properly characterize the narrower input pulses. If this is the case, then the position of the curves in Graphs PTP01A, PFN01A, PTP02A and PFN02A at the corner frequency might be altered by using a shorter sample interval. Conversely, Graphs PFT01A and PFT02A would likely exhibit a greater error

at the corner frequency because the curves would be shifted upward. The results discussed in Section 6.2.5 provide further comment on the effect of the sample interval. In future work an investigation should be conducted to determine the effect the pulse curve sample interval has on the position of the amplitude ratio and phase lag curves

6.2.3 The Effect of Spectral Smoothing on the Amplitude Ratio, Phase Lag and Frequency Content

Spectral smoothing was implemented using the formulae outlined in Section 3.4 using the program SMOTH discussed in Section 5.4.2.4. The investigation into spectral smoothing consisted of a series of trials involving application of the Hamming and Hanning spectral smoothing formulae. The author is not familiar with the detailed mathematical basis of the filters and it is not known if other filters, such as the exponential or the least squares polynomial outlined in Section 5.4.2.2 would not have produced similar results. Nonetheless interesting results were obtained by applying the filters and possibly further studies will yield more significant information.

In general spectral smoothing helped reduced the "noise" that appeared on some amplitude ratio plots due to the zeros in the frequency content curve of the input pulse (Section 6.2.1.2). The effectiveness of spectral smoothing increased as the input pulse duration increased. The graphs derived from runs 01, 05 and 06 presented in Section 6.2.1.3 demonstrate how the effect of spectral smoothing increases with increasing pulse duration. (Note: In some cases, identical plots appear sequentially due to an error in the smoothing program which is explained

in Section 6.2.3.1). The effectiveness of the Hanning smoothing method was greater than that of the Hamming method because it involved the application of two formulae in succession opposed to the single formula for the Hamming method. The cases presented in Section 6.2.3.1 for runs 01, 05 and 06 clearly demonstrate the relative effects of each filtering method. Although the majority of the graphs presented in Section 6.2.3.1 were derived using the FFT, a few graphs were derived using the FN technique and are presented to show that the effect was similar in this case.

It was observed that the zeros of the frequency content curve appeared to be removed after spectral smoothing was applied. Since it is these zeros that cause the noise in the amplitude ratio, it is reasonable to assume that when noise is reduced by filtering, the zeros in the associated frequency content curve should be removed. Graphs PFT06C, PFT06G and PFT06J in Section 6.2.3.1 demonstrate the effect of spectral smoothing on the frequency content curve.

A large number of graphs are presented in Section 6.2.3.1 so that anyone doing further work in this area may use them as a reference. It is difficult to comment here on all the comparisons which might be made so the reader is asked to use the table preceding the graphs and graph labelling code outlined in Section 6.1 to make any comparisons desired.

It is interesting to note that the frequency at which the rising tail in Graphs PFT06N and PFT06R and the falling tail in Graphs PFT06O and PFT06S begins is the frequency at which aliasing would be predicted by the sampling theorem. Aliasing is predicted when $w = \pi/\Delta t$; for run 06 Δt was 0.2 so aliasing should occur at $w =$ 15.7 radians per second. This agrees very closely with the frequency at which the tails

on the graphs referred to above rise and fall.

6.2.3.1 Amplitude Ratio, Phase Lag and Frequency Content Graphs Illustrating the Effect of Spectral Smoothing

General comments:

a. Table 3 describes the graphs presented on pages 142 - 164 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs was given in Section 6.1.

c. Data presented on these graphs were derived from the time domain pulse data presented in Appendix I. Two Fourier transform techniques were used in the derivation:

   1. Fast Fourier transform (FT)

   2. Filon's quadrature (TP)

d. Only two phase lag graphs are shown; Graphs PFT06S and PFT06O. Breakdown of subroutine ANGLE (Section 5.4.2.5) rendered most phase lag graphs meaningless

e. Discontinuities and noise which is evident in the graphs is due to the zeros in frequency content curves of the corresponding input pulses.

f. Similar graphs are not necessarily on the same scale so careful consideration must be given to the scales when comparisons are made.

g. The smoothing program contained a minor error which resulted in the loss of a data point. Where two graphs with identical identifications appear simultaneously, the first is derived from

the corrected version of the smoothing routine and the second is derived from the erroneous version. <u>When graphs of smoothed data do not appear in identical copies, they have been derived using the erroneous smoothing routine; however, they still serve to illustrate the effect of smoothing</u>.

h. The frequency content graphs presented on pages 162 - 164 apply to the input pulses shown in Appendix I. They demonstrate the effect of spectral smoothing on the frequency content curve.

i. For information on the sample interval used on the time data associated with these runs, see Section 6.2.1.3.

j. The theoretical high-frequency asymptote for the amplitude ratio curve has been drawn on the graphs along with a "+" at the theoretical position of the amplitude ratio curve at the corner frequency. Also, on the phase lag graphs a "+" appears at the theoretical position of the phase lag curve at the corner frequency.

k. The graphs on pages 142 - 164 are presented in the following general order:

   1. amplitude ratio calculated by FT presented in order of increasing amounts of smoothing and in order of increasing input pulse width.

   2 phase lag calculated by FT in the same order as stated in 1.

   3. frequency content calculated by FT in the same order as stated in 1.

4. amplitude ratio calculated by FN in the same order as stated in 1.

5. frequency content calculated by FN in the same order as stated in 1.

6.2.4 The Effect of an Output Pulse Error of Closure on the Amplitude Ratio With and Without Spectral Smoothing

A limited number of analyses were performed with an error of closure created by using the technique outlined under item f of Section 6.2.4.1. The results obtained agree with those reported by Clements (12), Dreifke and Hougen (24) and Hougen and Walsh (42) in that truncating the tail of the output pulse and creating an error of closure has a detrimental effect on the accuracy of the amplitude ratio curve obtained from the pulse data. In this work a general increased scatter occurred in amplitude ratio data at high frequencies when the error of closure was largest. The workers (12), (24) and (42) also observed that an error of closure in the output pulse caused the amplitude ratio curve to appear steeper than it should. They reported that in the presence of a large error of closure the results of a first order system could be interpreted as second order behaviour. In the case of Graph PFN03W, in Section 6.2.3.1, the advanced position of the curve relative to the theoretical corner frequency and the steep initial slope after the corner frequency could be interpreted as exhibiting this type of behaviour.

In the author's opinion the most significant effect of the error of closure was the obscuration of the discontinuities in the amplitude ratio curve which are due to the zeros in the frequency content

TABLE 3

LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF SPECTRAL SMOOTHING

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT01K	142	1.0	FT	Amplitude ratio; no smoothing
PFT01N	143	1.0	FT	Amplitude ratio; Hamming smoothing. Compare to PFT01K
PFT01R	144	1.0	FT	Amplitude ratio; Hanning smoothing. Compare to PFT01K
PFT05K	145	25.0	FT	Amplitude ratio; no smoothing
PFT05N	146	25.0	FT	Amplitude ratio; Hamming smoothing. Ordinate scale change compared to PFT05K.
PFT05R	147	25.0	FT	Amplitude ratio; Hanning smoothing. Ordinate scale change compared to PFT05K.
PFT06K	148	75.0	FT	Amplitude ratio; no smoothing
PFT06N	149	75.0	FT	Amplitude ratio; Hamming smoothing. Ordinate scale change compared to PFT06K. Rising tail illustrates aliasing. Graph obtained by using the corrected smoothing program.

TABLE 3 (CONT'D)

LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF SPECTRAL SMOOTHING

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT06N	150	75.0	FT	Same graph as preceding but obtained by using the erroneous smoothing program.
PFT06R	151	75.0	FT	Amplitude ratio; Hanning smoothing Ordinate scale change compared to PFT06K. Rising tail illustrates aliasing. Graph obtained by using the corrected smoothing program.
PFT06R	152	75.0	FT	Same graph as above but obtained by using the erroneous smoothing program.
PFT06O	153	75.0	FT	Phase lag; Hamming smoothing. Derived from same analysis as PFT06N. Dropping tail illustrates aliasing. Graph obtained by using the corrected smoothing program.
PFT06O	154	75.0	FT	Same graph as above but obtained by using the erroneous smoothing program.

TABLE 3 (CONT'D)

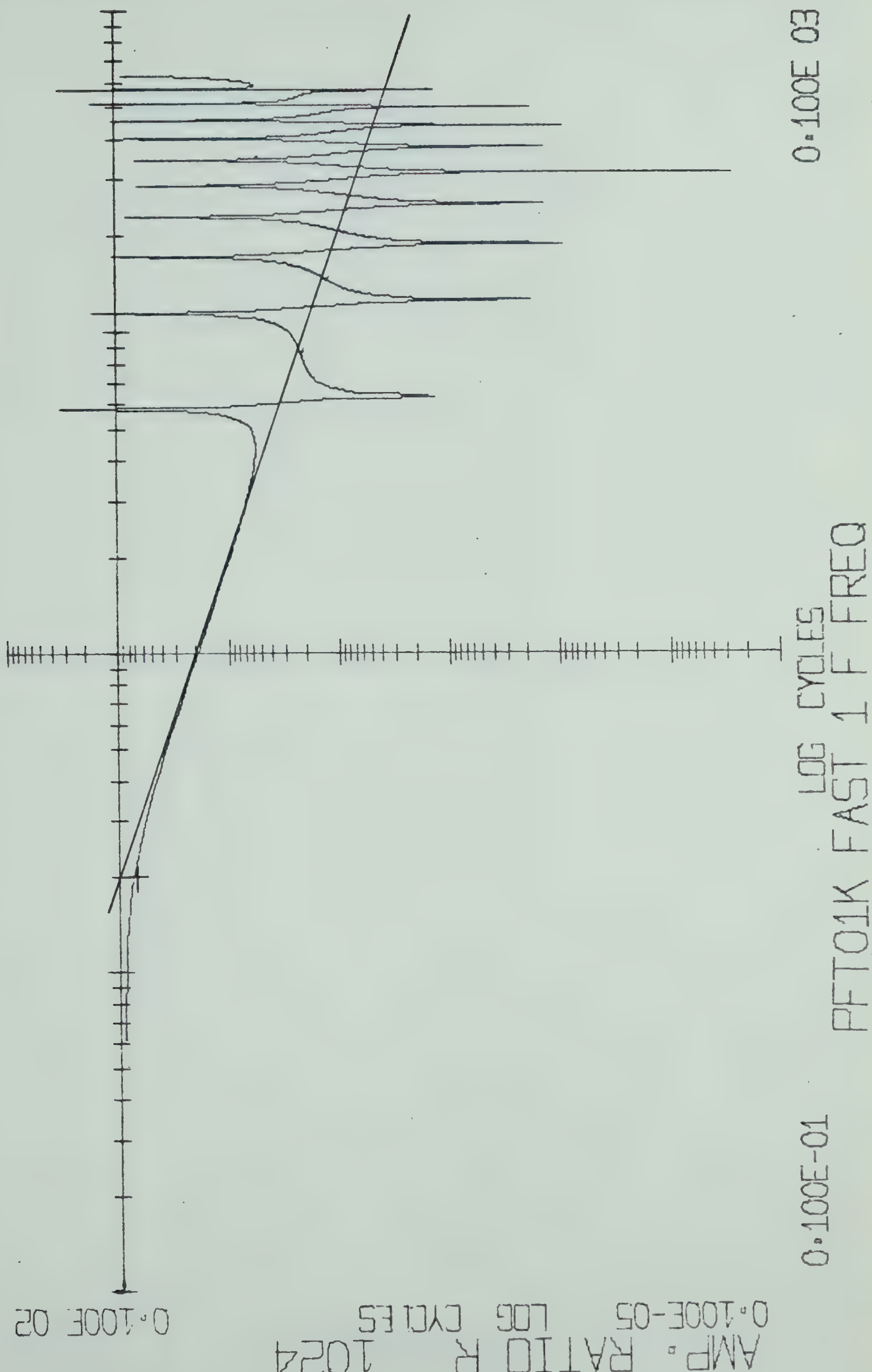
LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF SPECTRAL SMOOTHING

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT06S	155	75.0	FT	Phase lag; Hanning smoothing. Derived from same analysis as PFT06R. Dropping tail illustrates aliasing Graph obtained by using the corrected smoothing program.
PFT06S	156	75.0	FT	Same graph as above but obtained by using the erroneous smoothing program.
PFT06C	157	75.0	FT	Frequency content curve with no smoothing applied. (Note: The PFT06C here and in Appendix I are the same curve except for the frequency range plotted.)
PFT06G	158	75.0	FT	Frequency content curve for the same input pulse that yielded PFT06C but with Hamming smoothing applied.
PFT06J	159	75.0	FT	Frequency content curve for the same input pulse that yielded PFT06C but with Hanning smoothing applied.

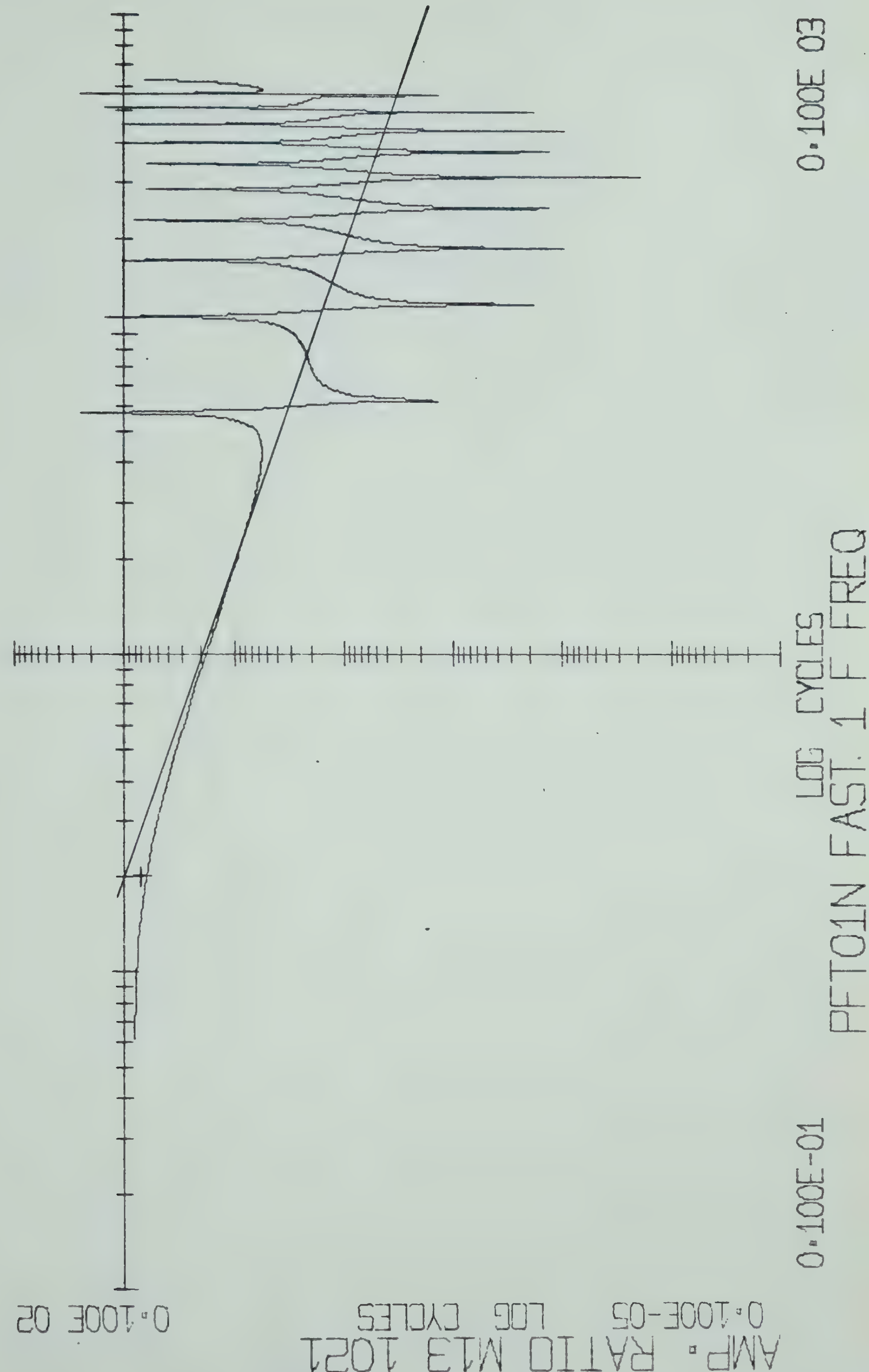
TABLE 3 (CONT'D)

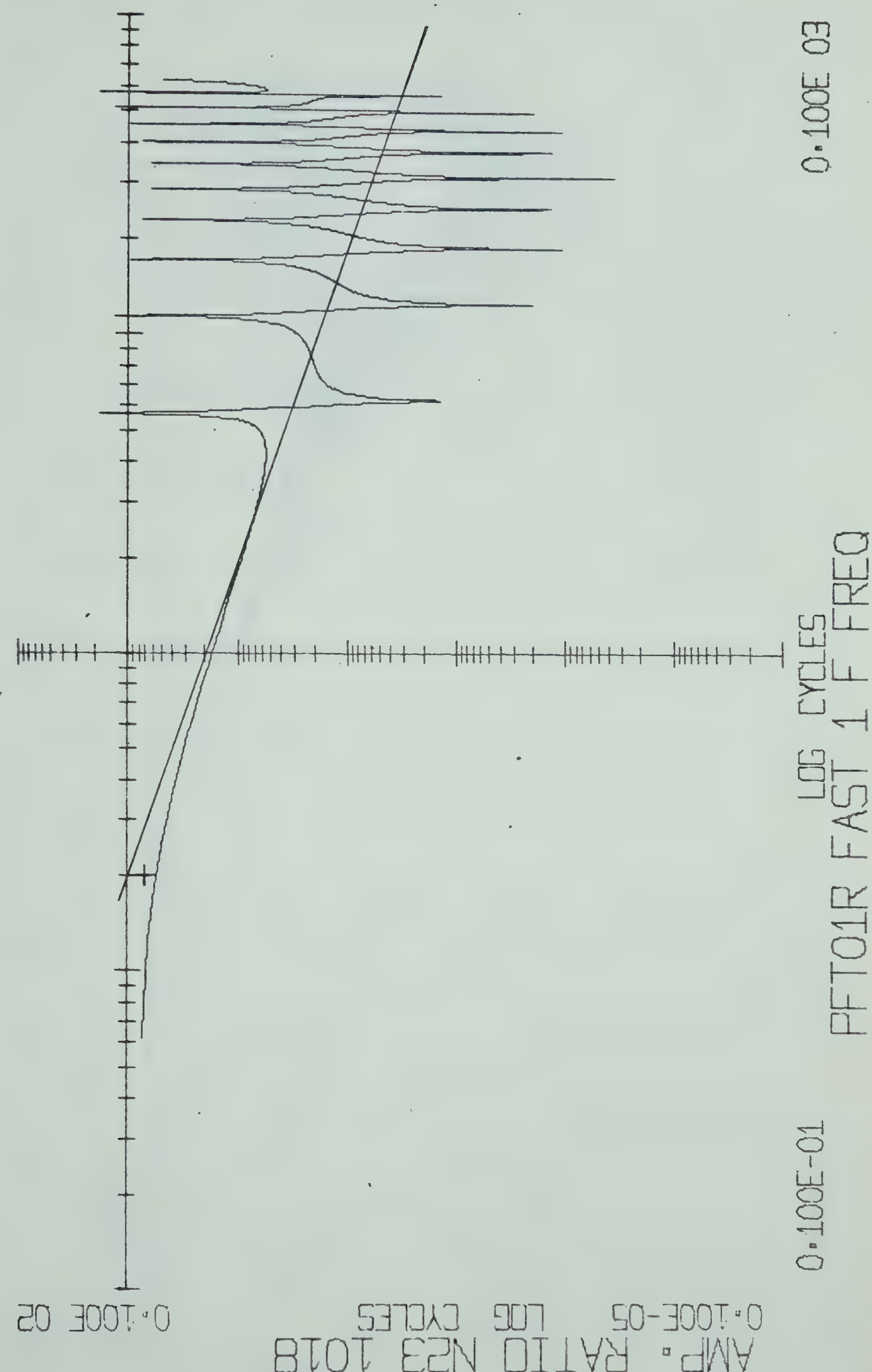
LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF SPECTRAL SMOOTHING

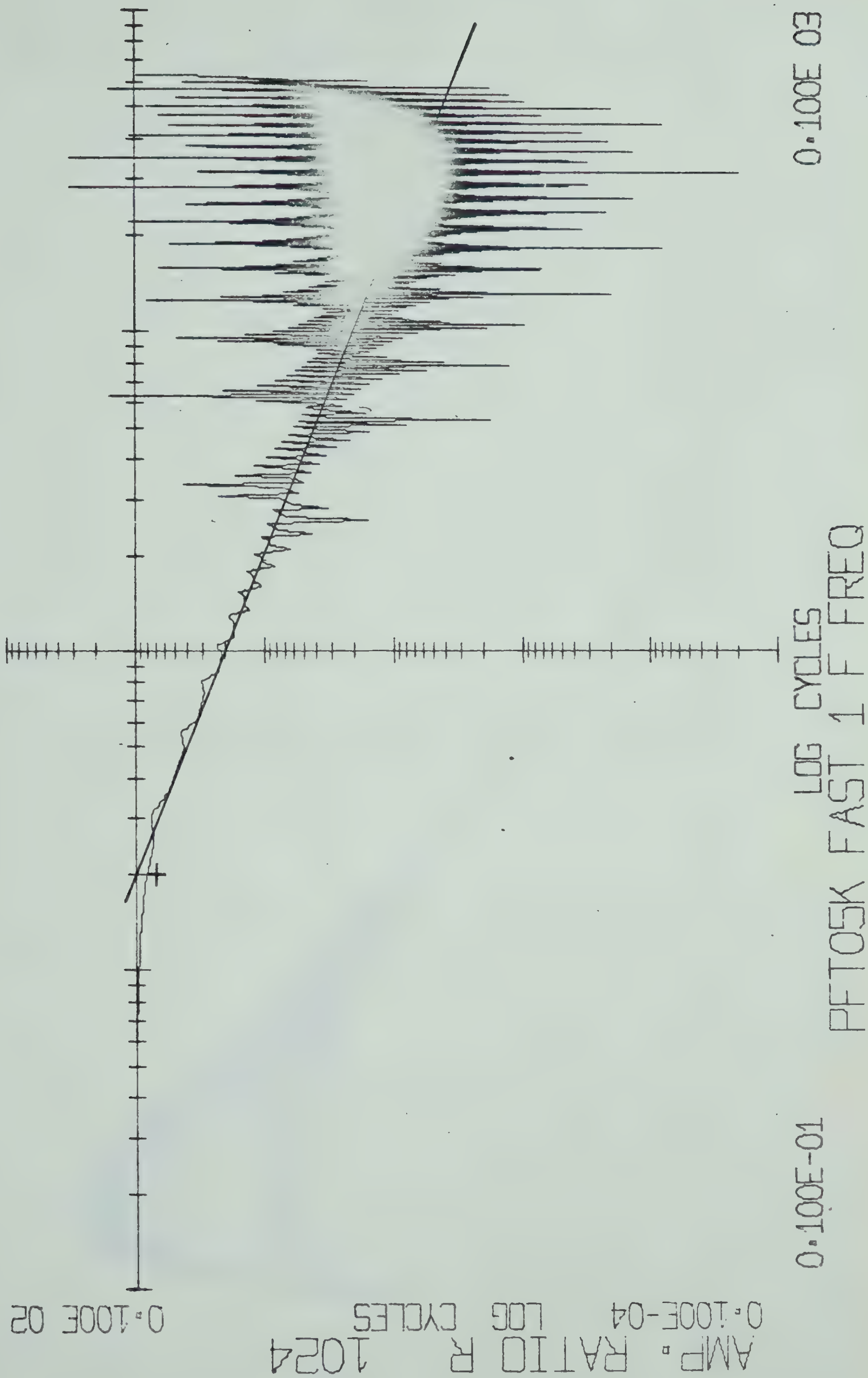
GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFN06E	160	75.0	FN	Amplitude ratio plot; equivalent to PFN06A (Section 6.2.1.3) but with Hamming smoothing applied.
PFN06H	161	75.0	FN	Amplitude ratio plot; equivalent to PFN06A (Section 6.2.1.3) but with Hanning smoothing applied.
PFN06C	162	75.0	FN	Frequency content curve for the input pulse in PTH06A in Appendix I. No smoothing applied.
PFN06G	163	75.0	FN	Frequency content curve for the same input pulse that yielded PFN06C but with Hamming smoothing applied.
PFN06J	164	75.0	FN	Frequency content curve for the same input pulse that yielded PFN06C but with Hanning smoothing applied.

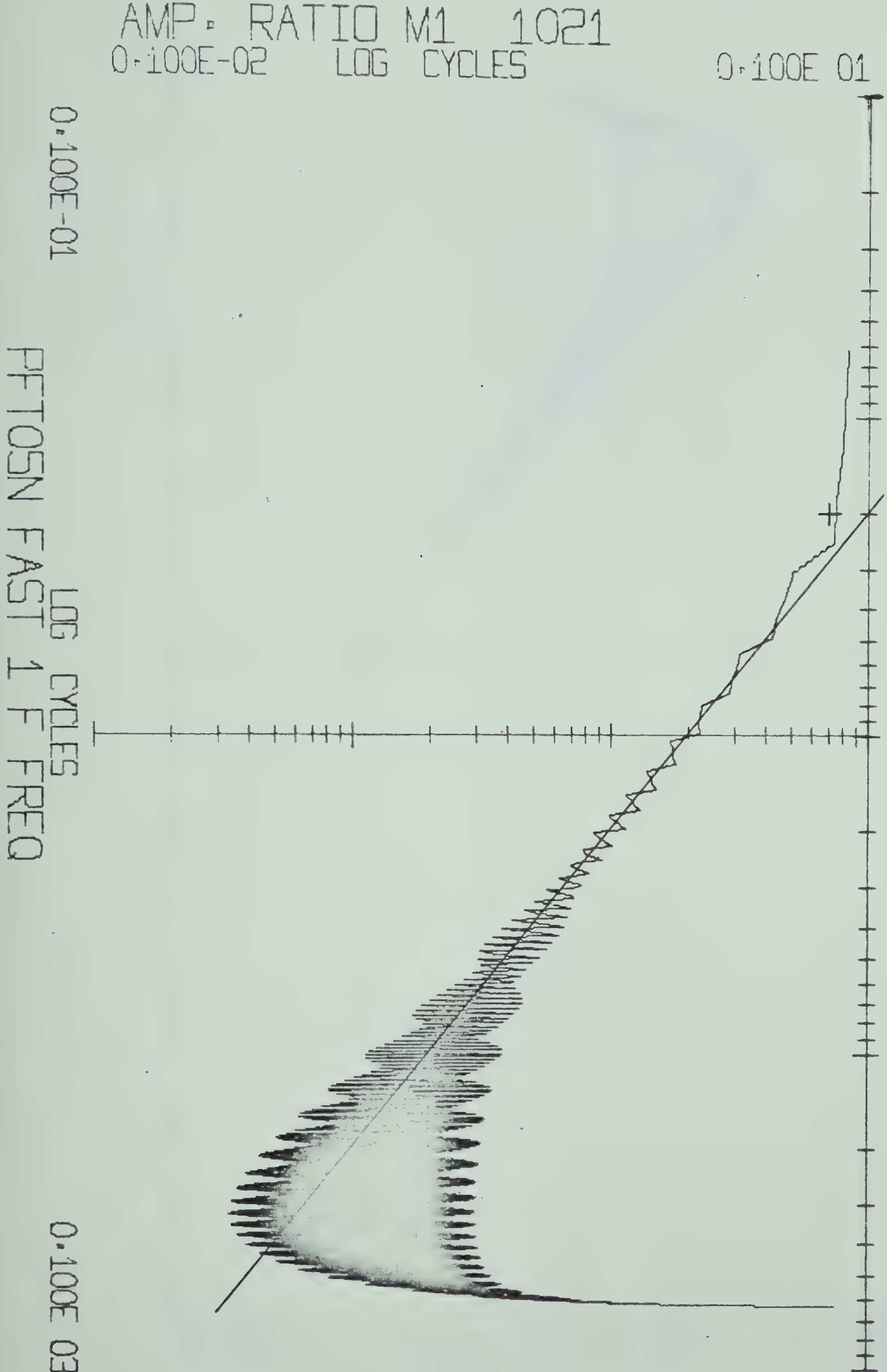


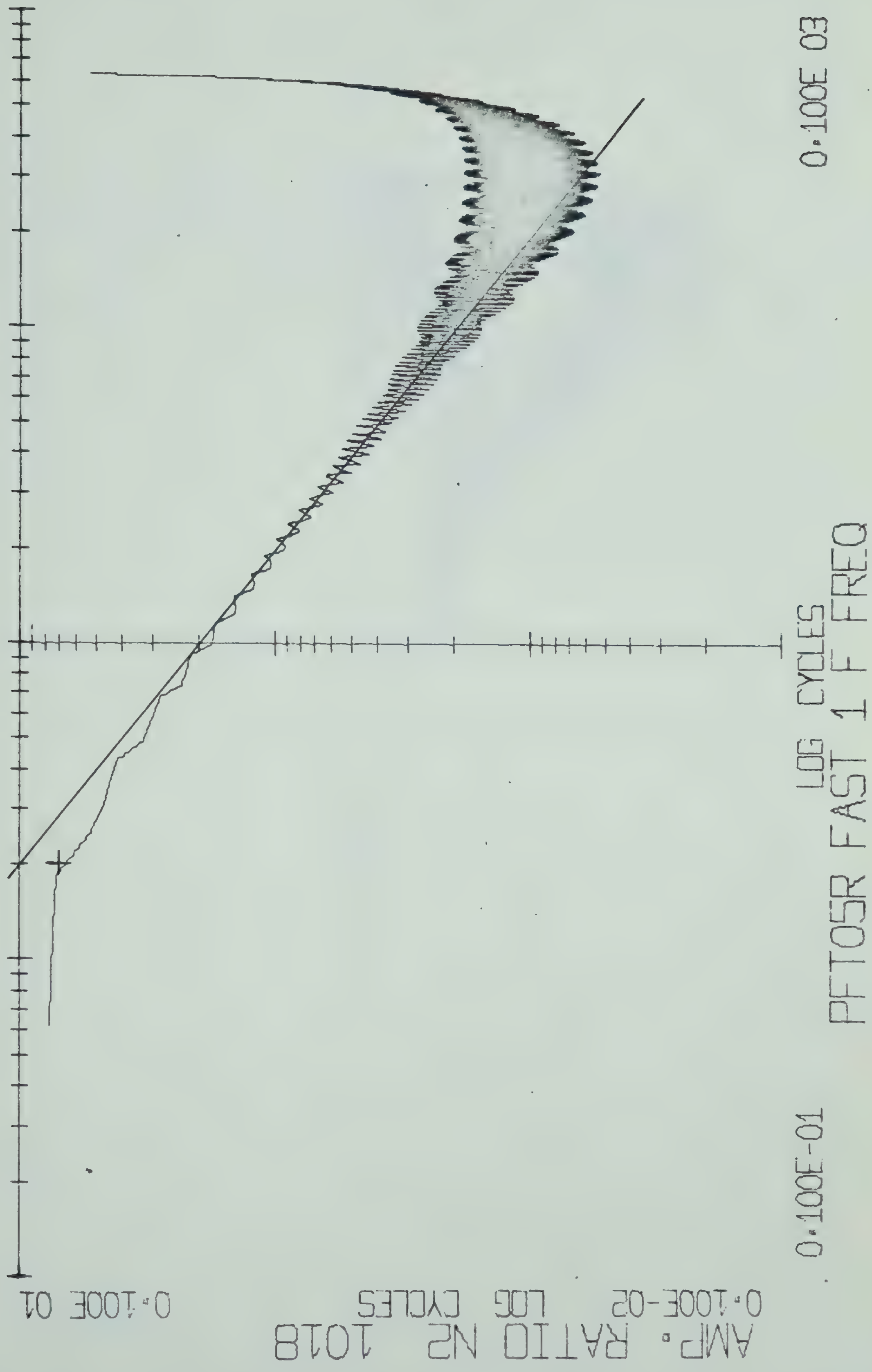
0.100E 03
LOG CYCLES
PFT01K FAST 1 F FREQ
0.100E-01
0.100E 02
0.100E-05 LOG CYCLES
AMP. RATIO R 1024

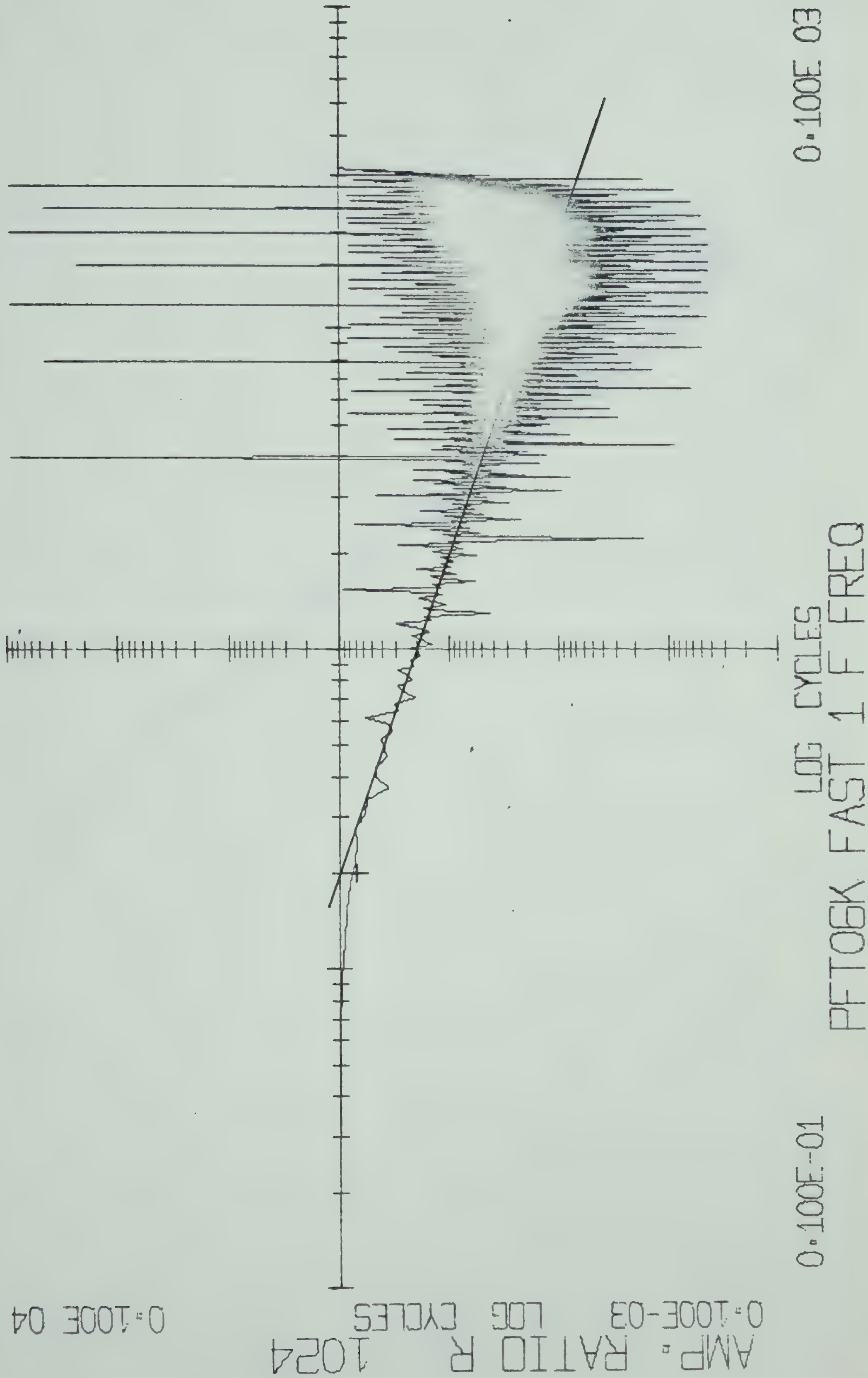

0.100E 03
0.100E-01
LOG CYCLES
PFT01N FAST 1 F FREQ
0.100E-05 LOG CYCLES
0.100E 02
AMP. RATIO M13 1021


0.100E 03
LOG CYCLES
PFT01R FAST 1 F FREQ
0.100E-01
0.100E 02
0.100E-05 LOG CYCLES
AMP. RATIO N23 1018

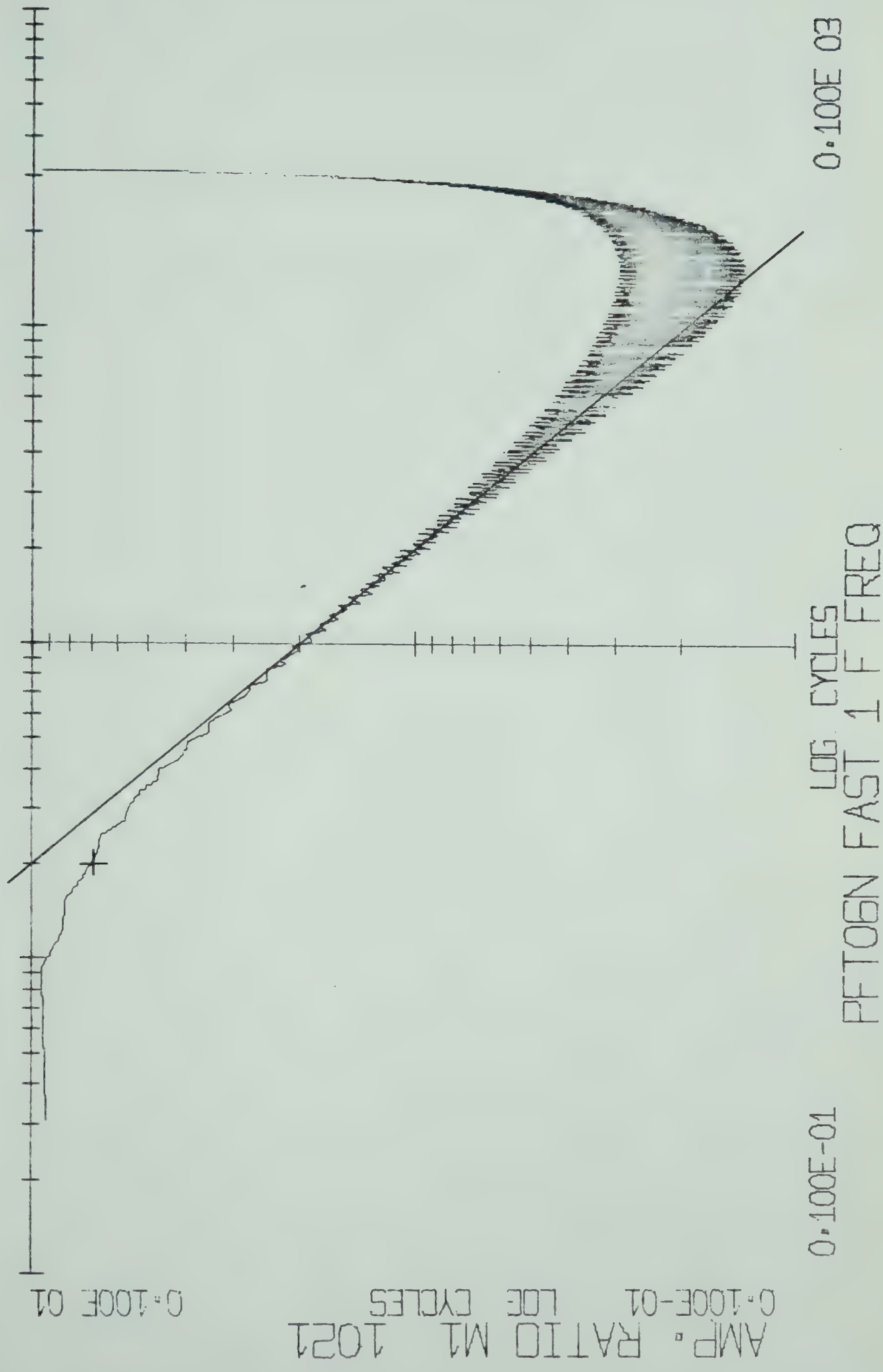

0.100E 03
LOG CYCLES
PFTO5K FAST 1 F FREQ
0.100E-01
0.100E 02
0.100E-04 LOG CYCLES
AMP. RATIO R 1024

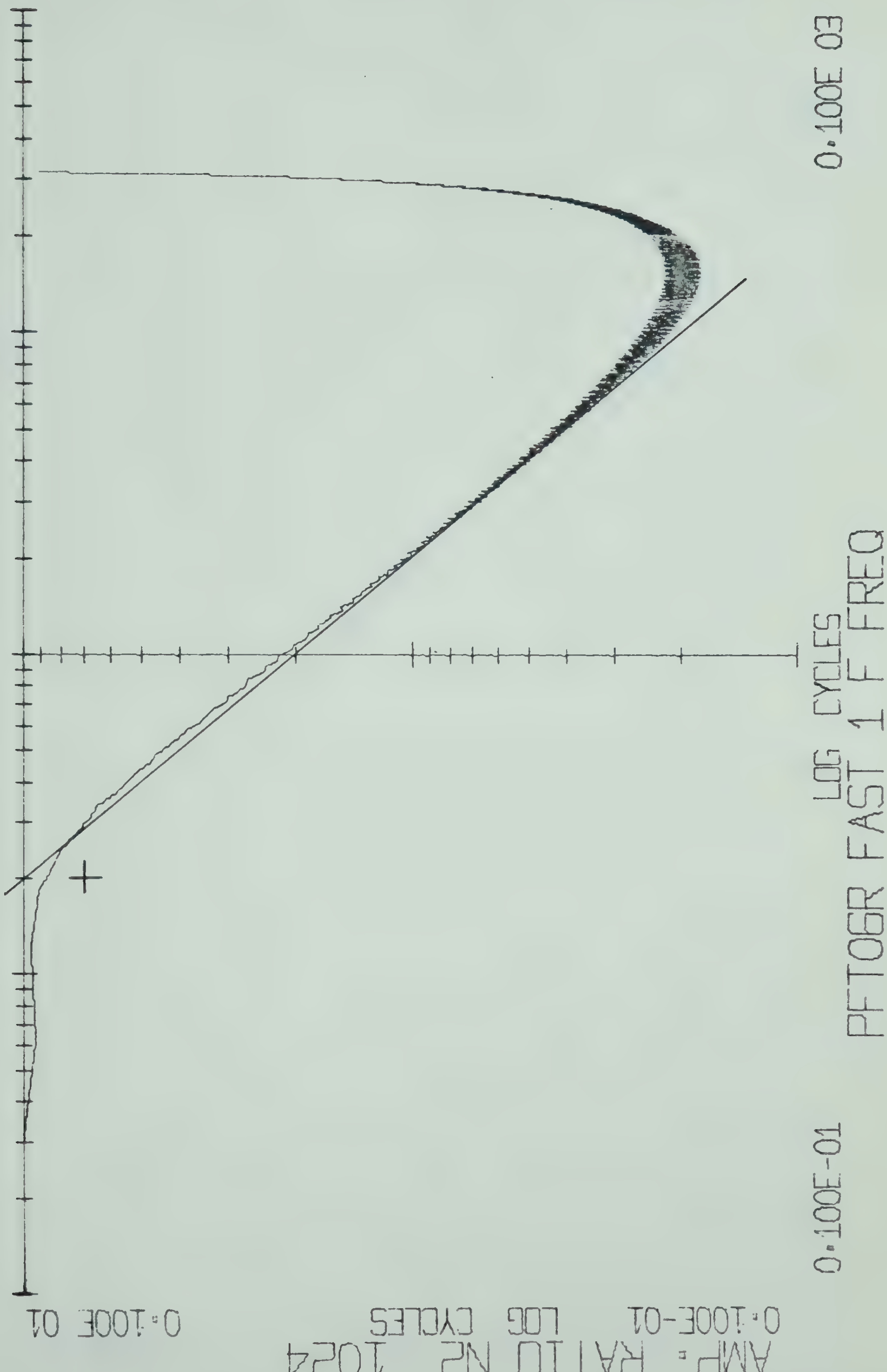

AMP. RATIO M1 1021
0.100E-02 LOG CYCLES 0.100E 01
0.100E-01
LOG CYCLES
0.100E 03
PFTO5N FAST 1 F FREQ

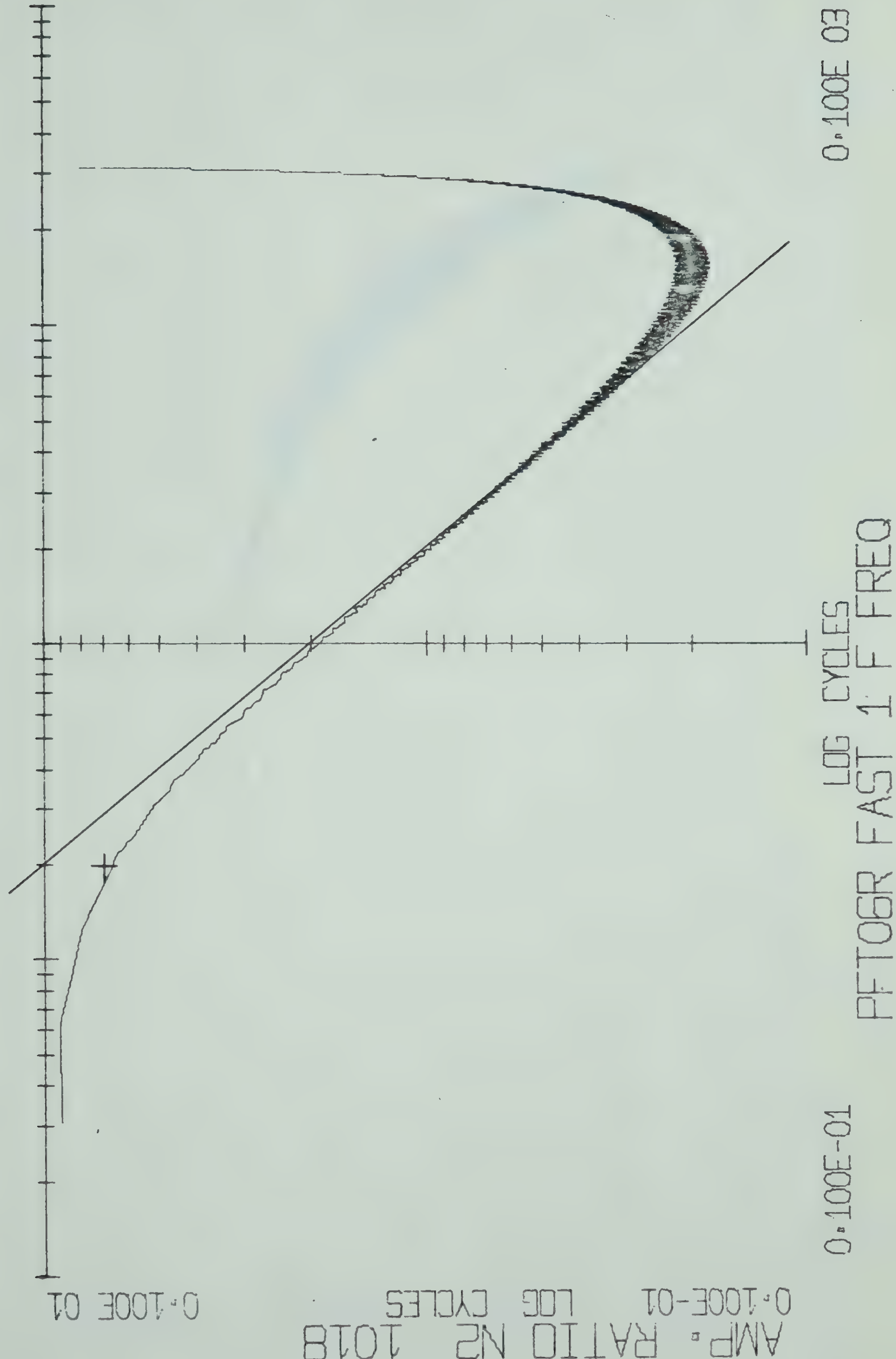

0.100E-01
0.100E 03
LOG CYCLES
PFT05R FAST 1 F FREQ
0.100E-02 LOG CYCLES
0.100E 01
AMP. RATIO N2 1018

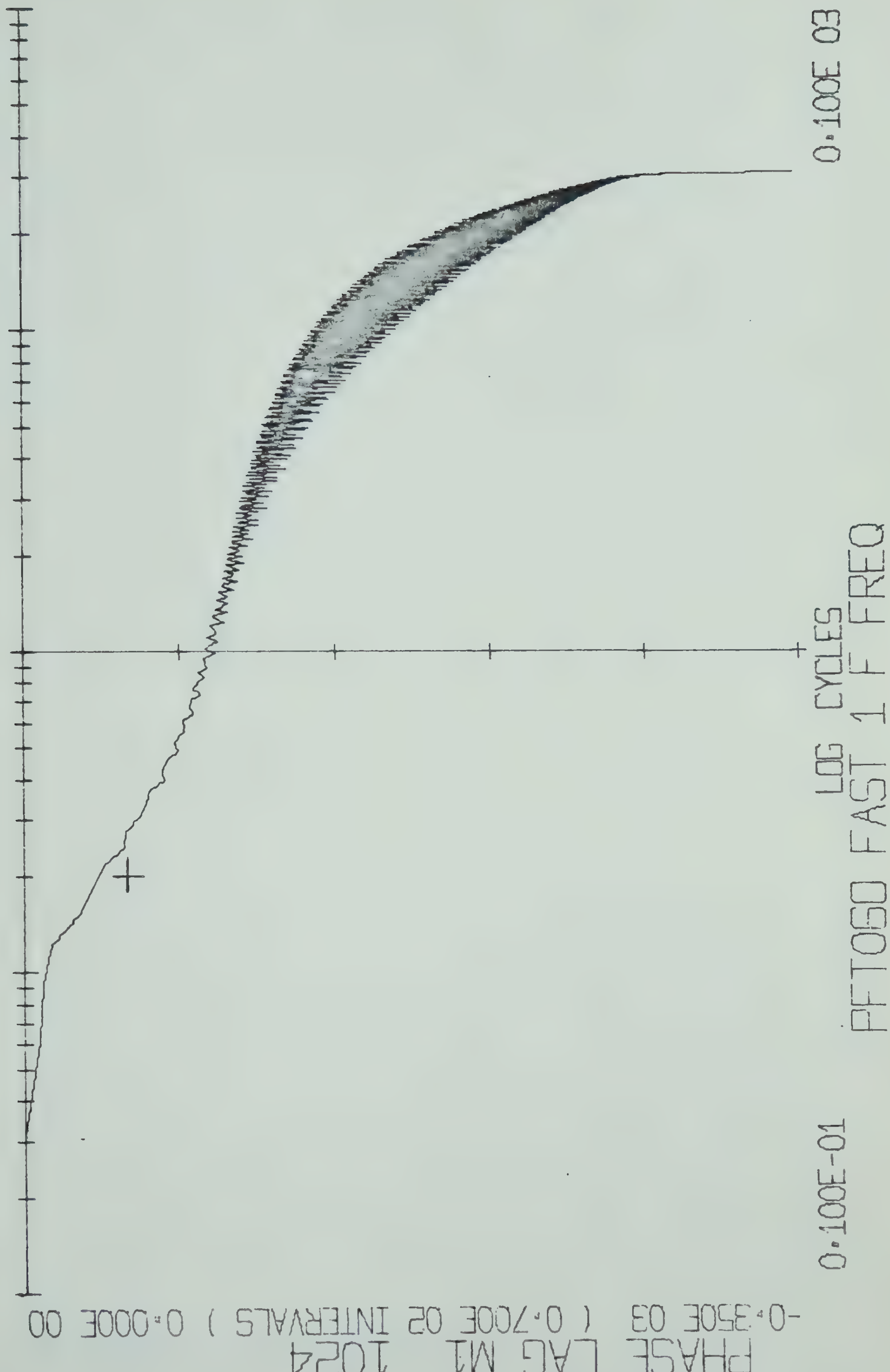

PFTO6K FAST 1 F FREQ
LOG CYCLES
0.100E-01
0.100E 03
AMP. RATIO R 1024
0.100E-03 LOG CYCLES
0.100E 04

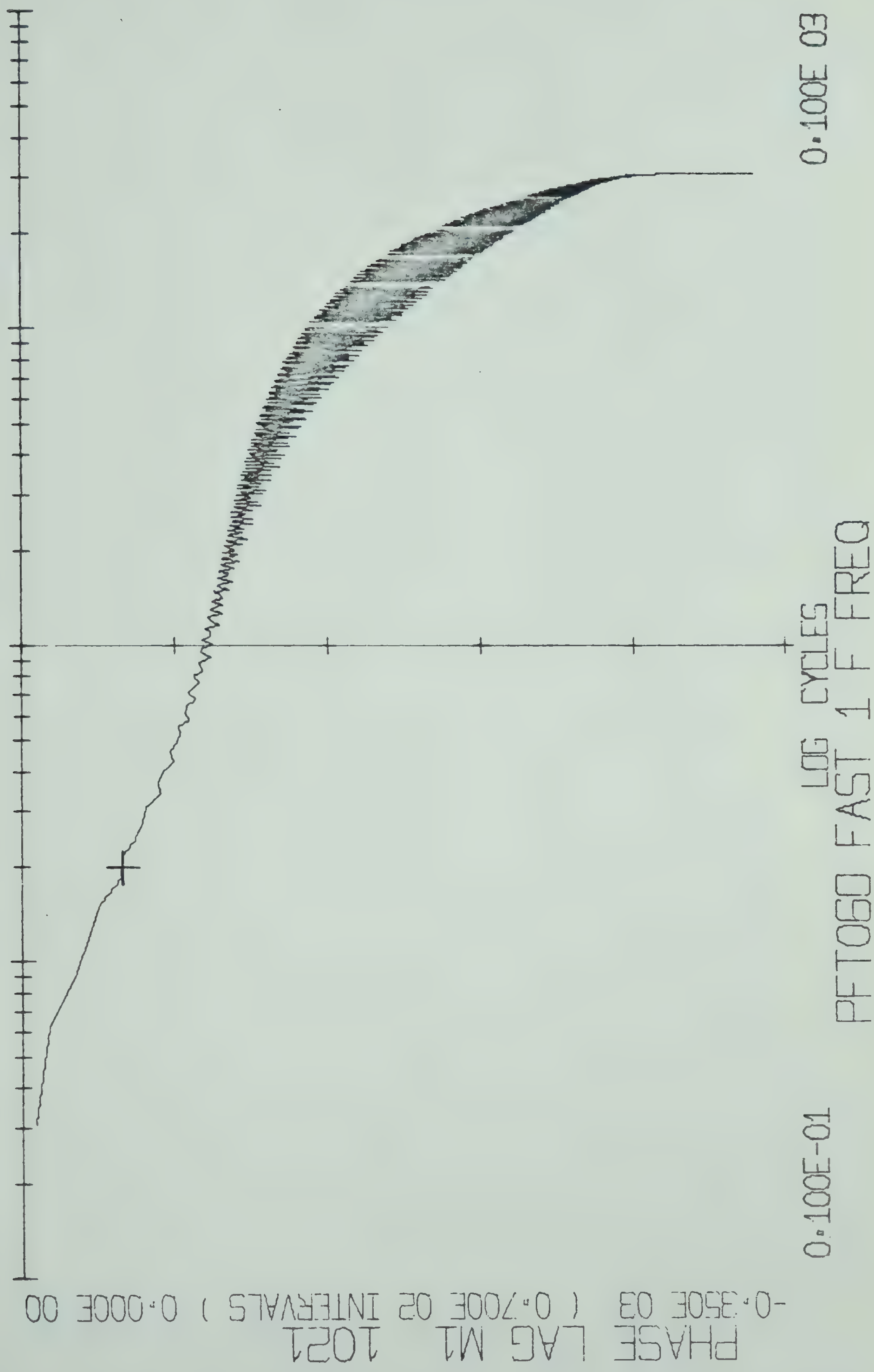

0.100E-01
0.100E 03
LOG CYCLES
PFTOGN FAST 1 F FREQ
0.100E-01
0.100E 01
LOG CYCLES
AMP. RATIO M1 1024

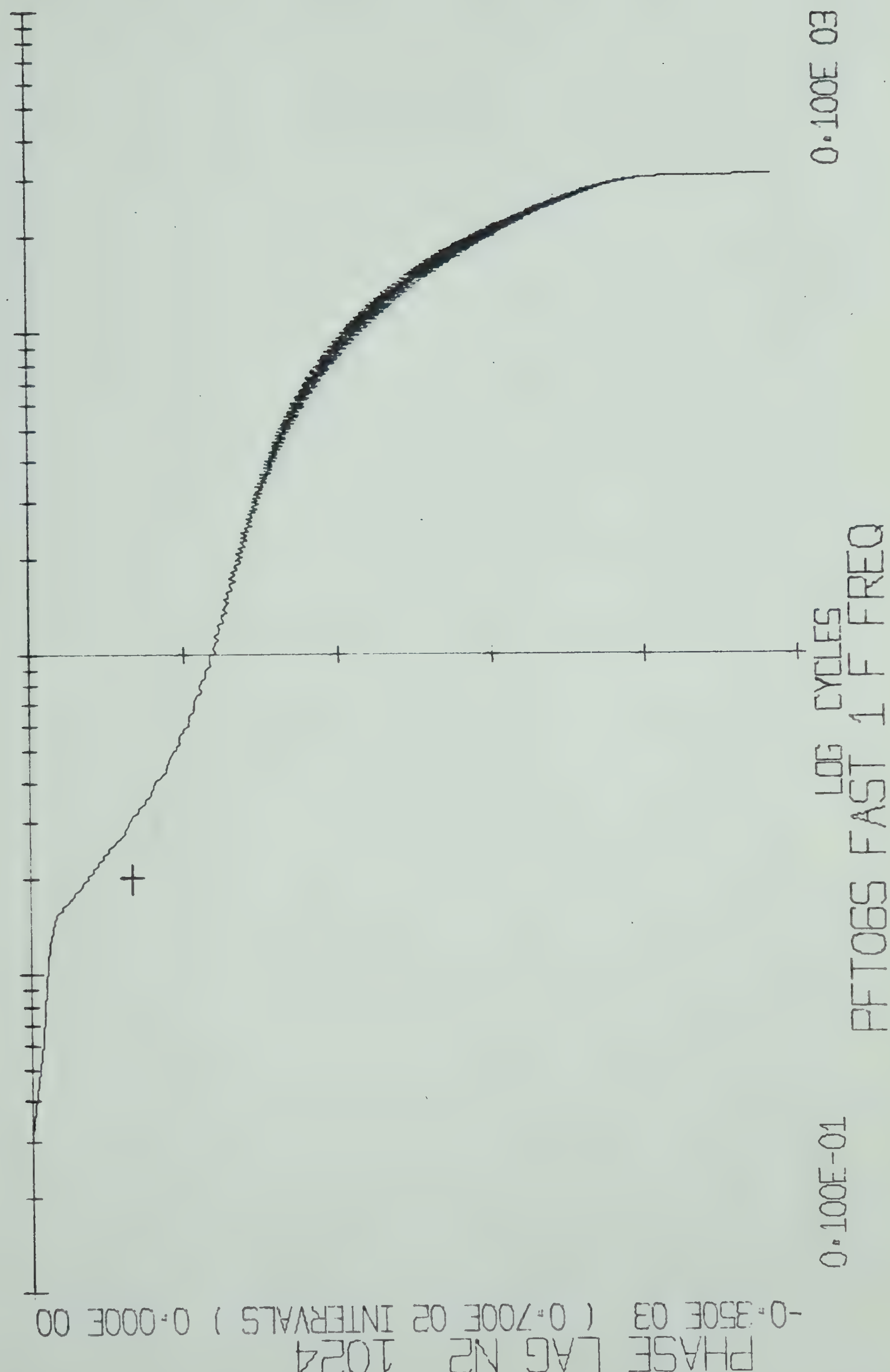

0.100E 03
LOG CYCLES
PFTOGN FAST 1 F FREQ
0.100E-01
0.100E 01
LOG CYCLES
0.100E-01
AMP. RATIO M1 1021

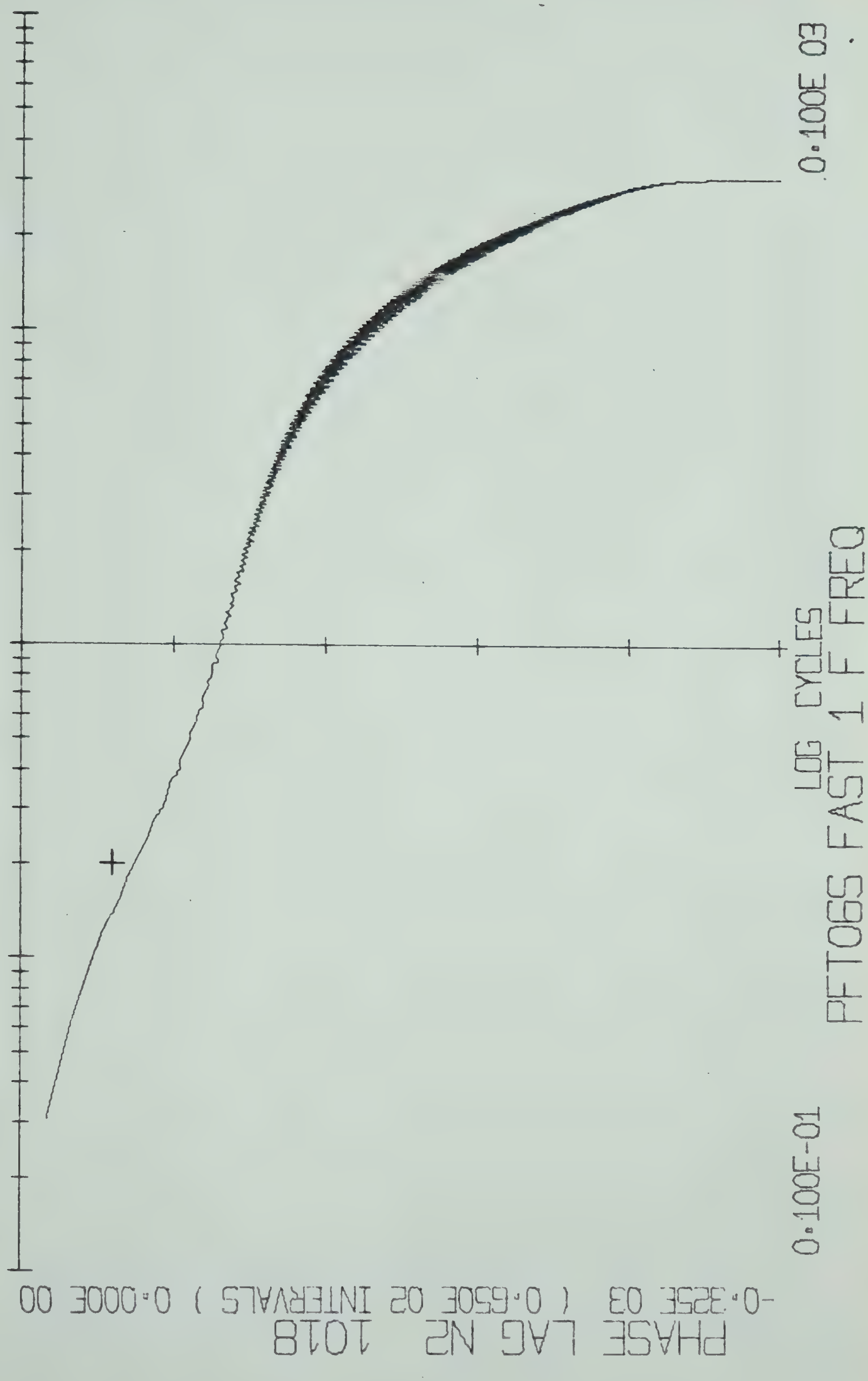

0.100E-01
0.100E 03
LOG CYCLES
PFTOGR FAST 1 F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO N2 1024

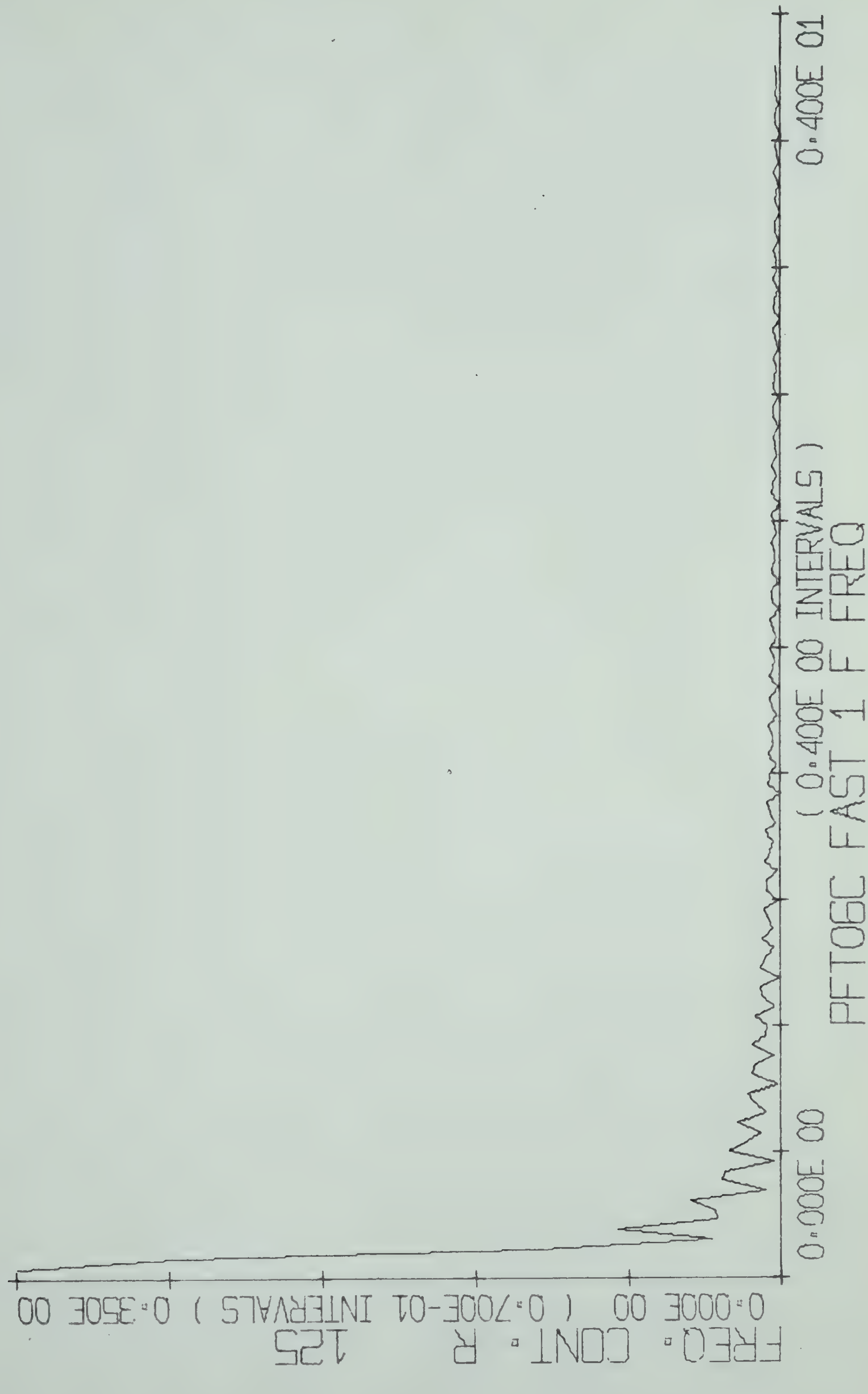

0.100E 03
LOG CYCLES
PFTOGR FAST 1 F FREQ
0.100E-01
0.100E 01
LOG CYCLES
0.100E-01
AMP. RATIO N2 1018


0.100E 03
0.100E-01
LOG CYCLES
PFT060 FAST 1 F FREQ
-0.350E 03 (0.700E 02 INTERVALS) 0.000E 00
PHASE LAG M1 1024

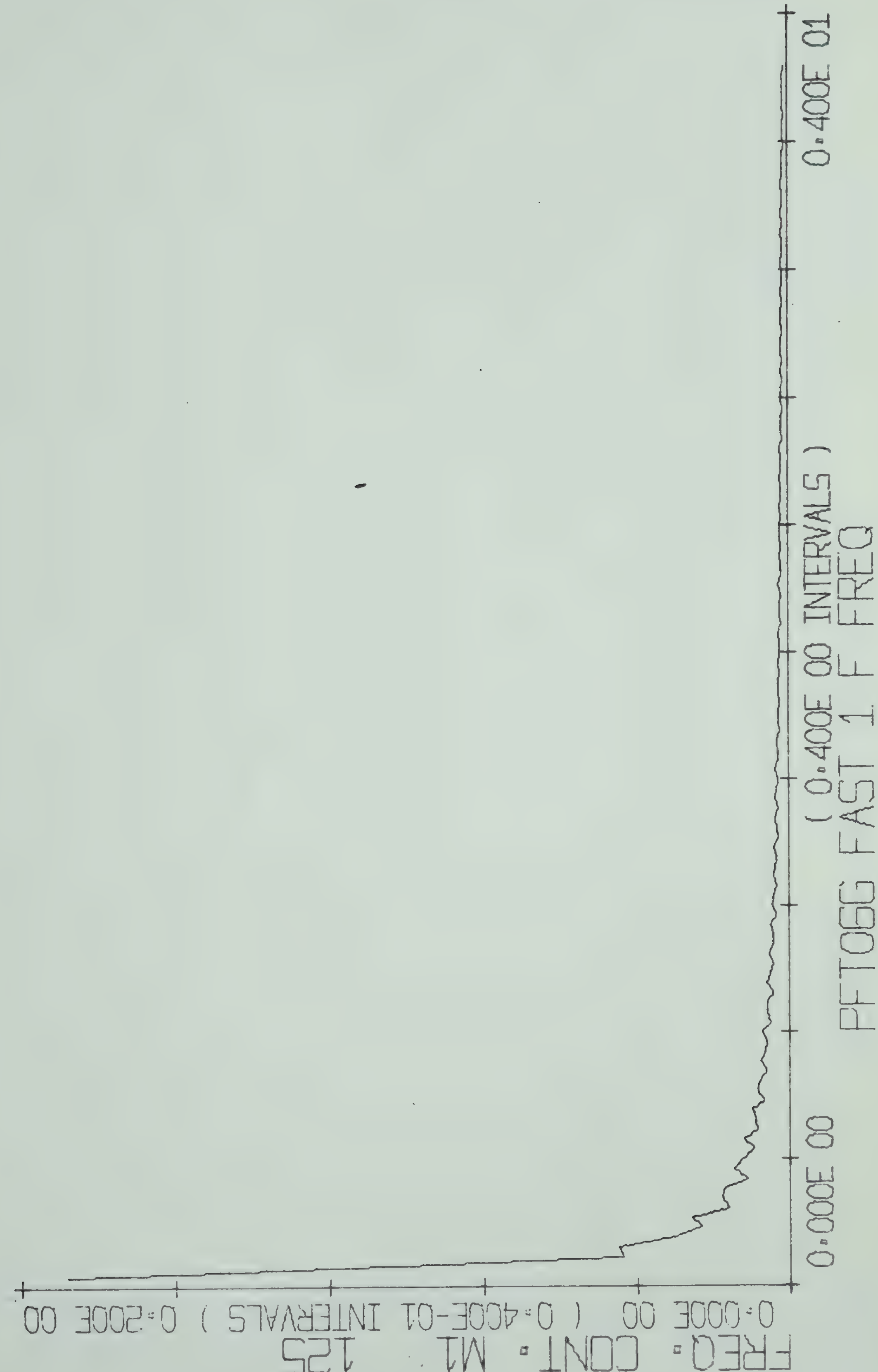

0.100E 03
LOG CYCLES
PFTO6D FAST 1 F FREQ
0.100E-01
-0.350E 03 (0.700E 02 INTERVALS) 0.000E 00
PHASE LAG M1 1021


0.100E 03
0.100E-01
LOG CYCLES
PFTO65 FAST 1 F FREQ
-0.350E 03 (0.700E 02 INTERVALS) 0.000E 00
PHASE LAG N2 1024

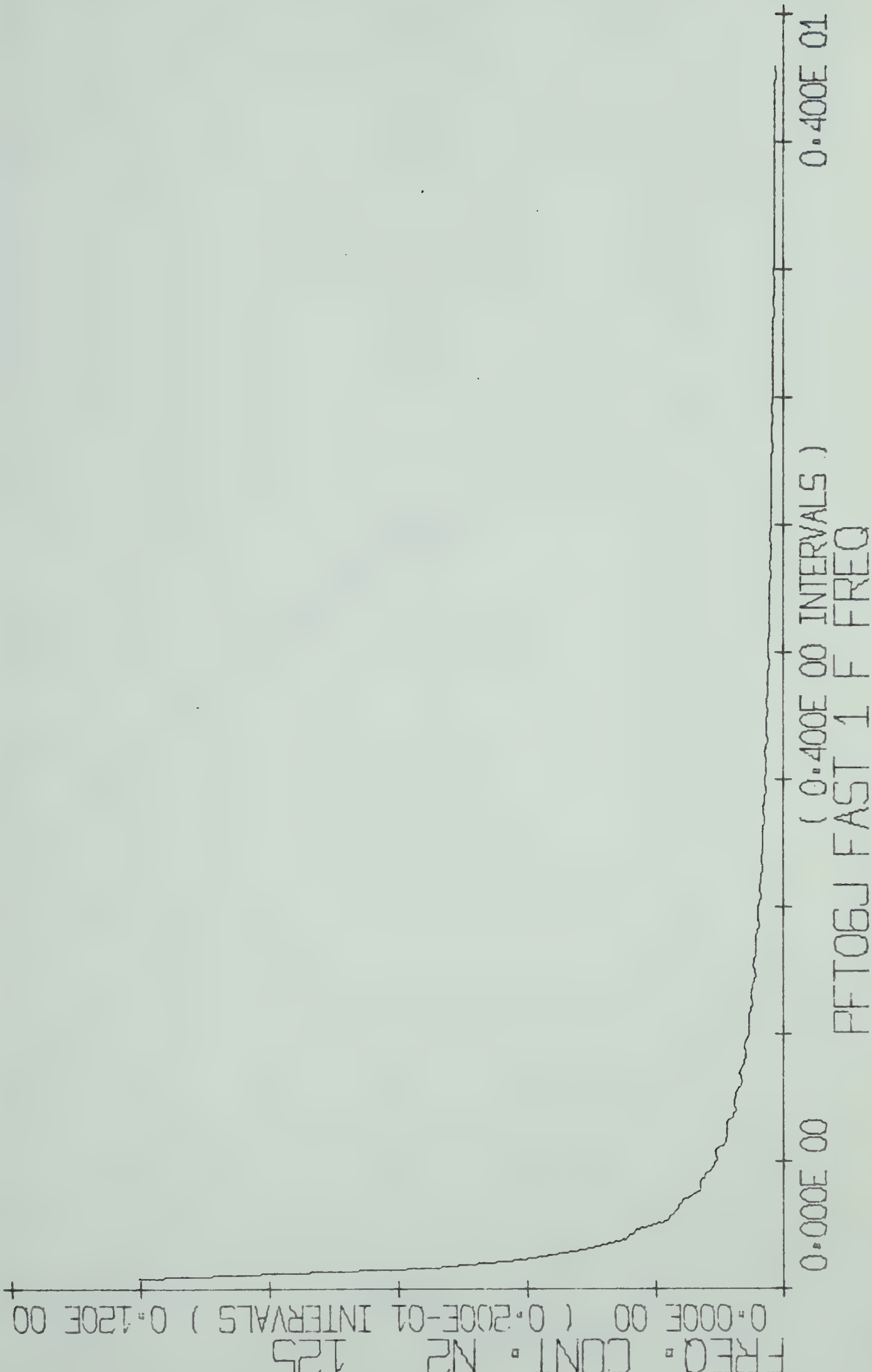

0.100E 03
LOG CYCLES
PFTOG5 FAST 1 F FREQ
0.100E-01
-0.325E 03 (0.650E 02 INTERVALS) 0.000E 00
PHASE LAG N2 1018

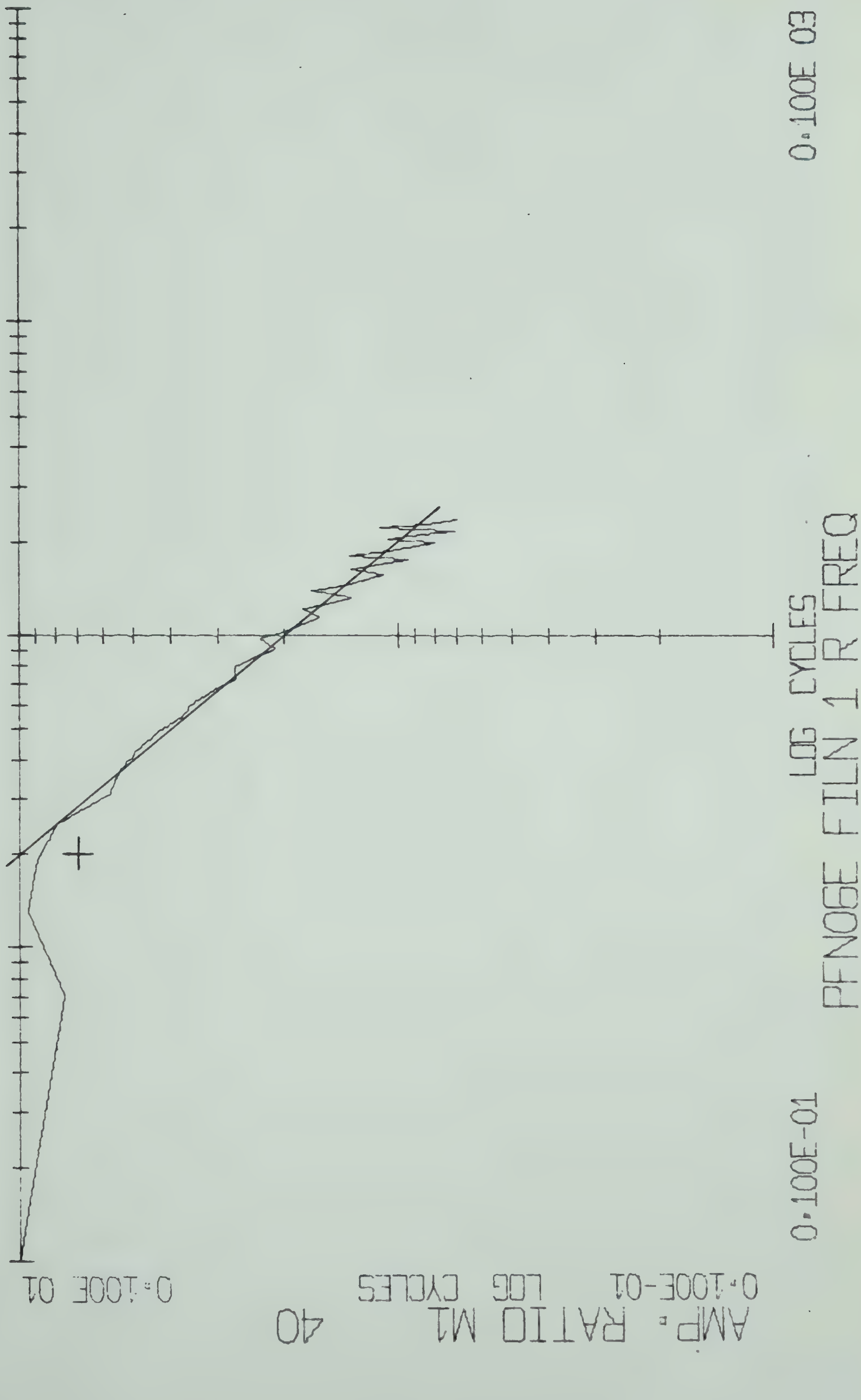


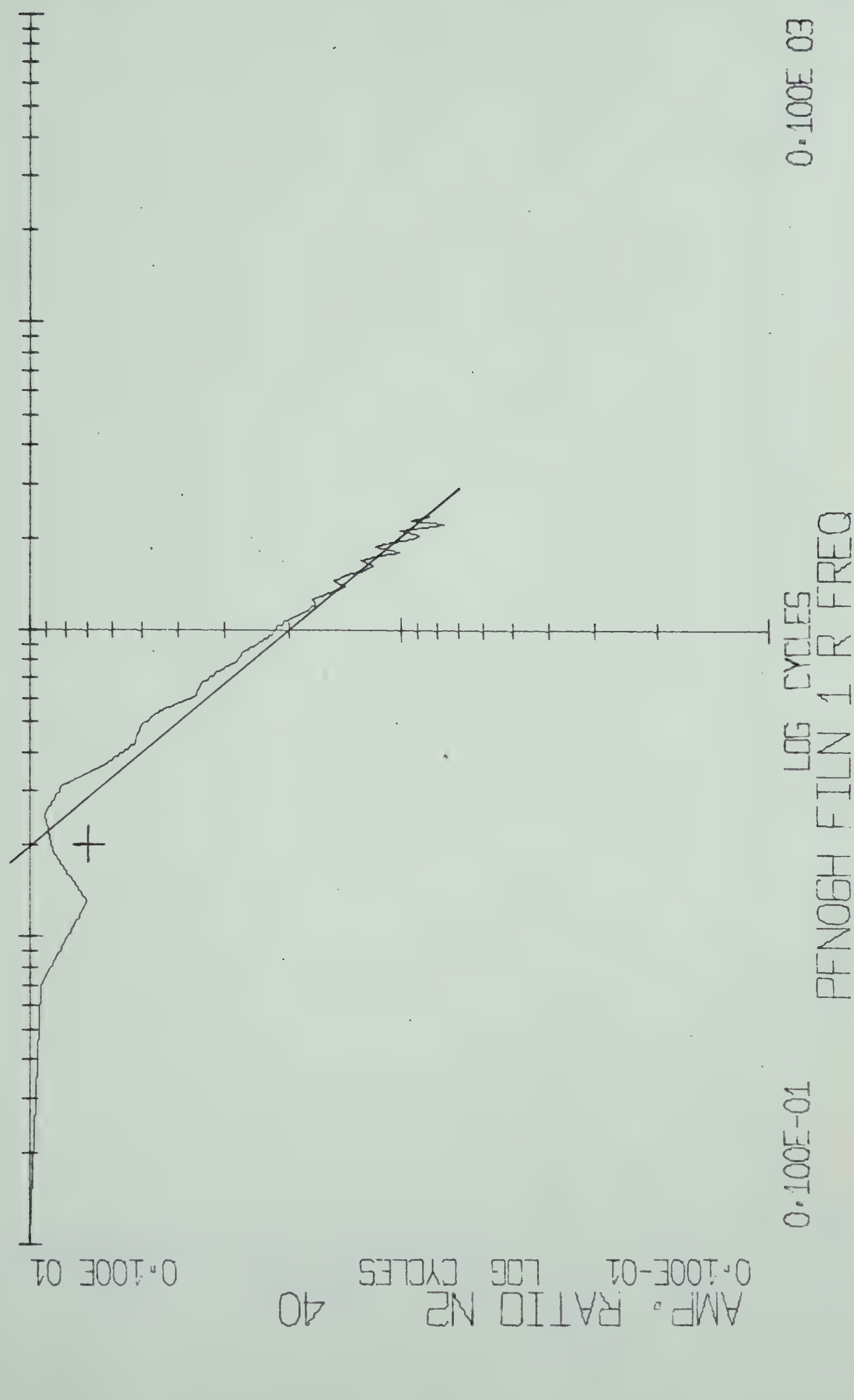
0.000E 00
(0.400E 00 INTERVALS)
0.400E 01
PFT06C FAST 1 F FREQ
FREQ. CONT. R 125
0.000E 00 (0.700E-01 INTERVALS) 0.350E 00

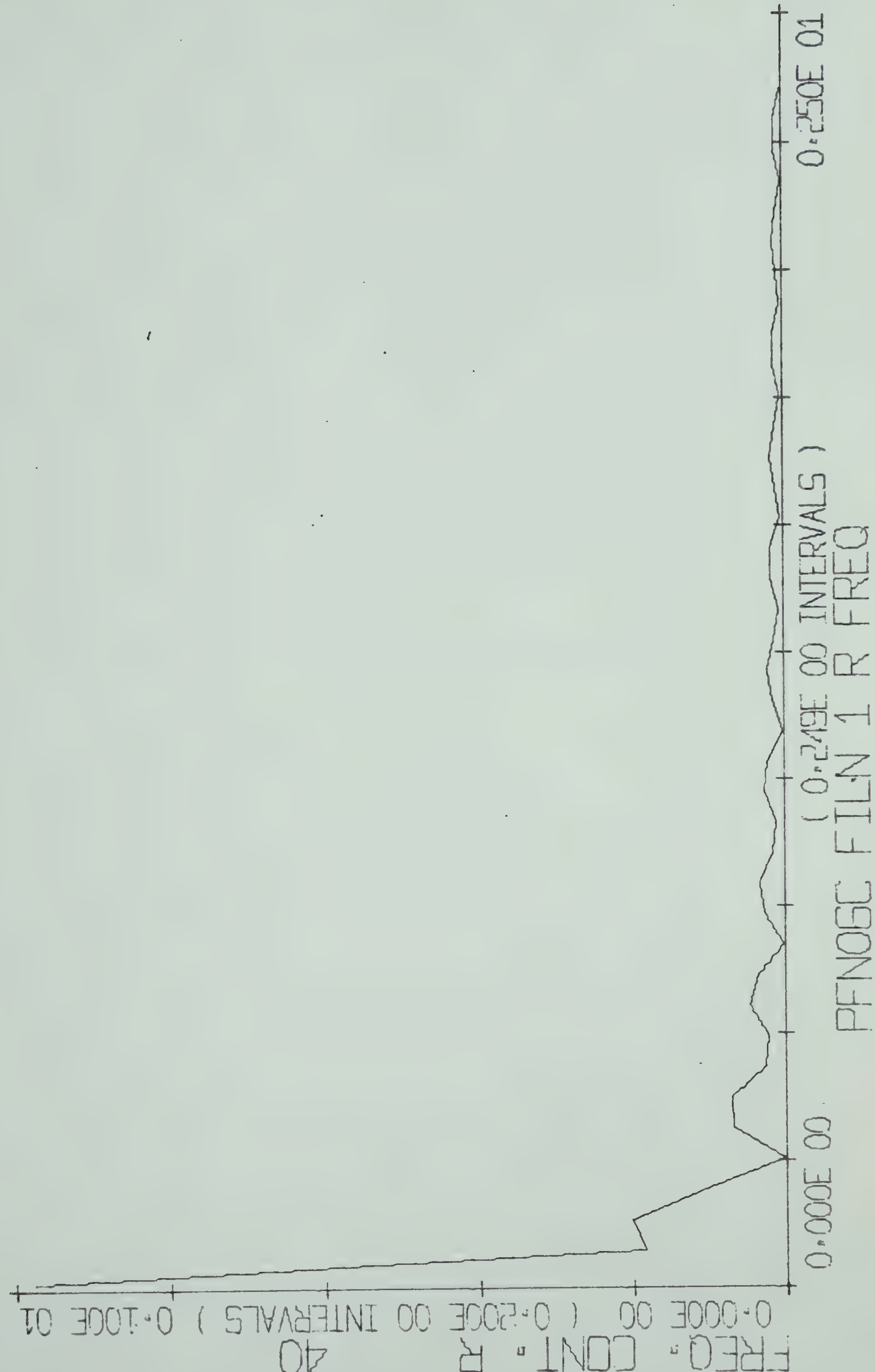


0.000E 00
0.400E 01
(0.400E 00 INTERVALS)
PFTOGG FAST 1 F FREQ
0.000E 00 (0.400E-01 INTERVALS) 0.200E 00
FREQ. CONT. M1 125

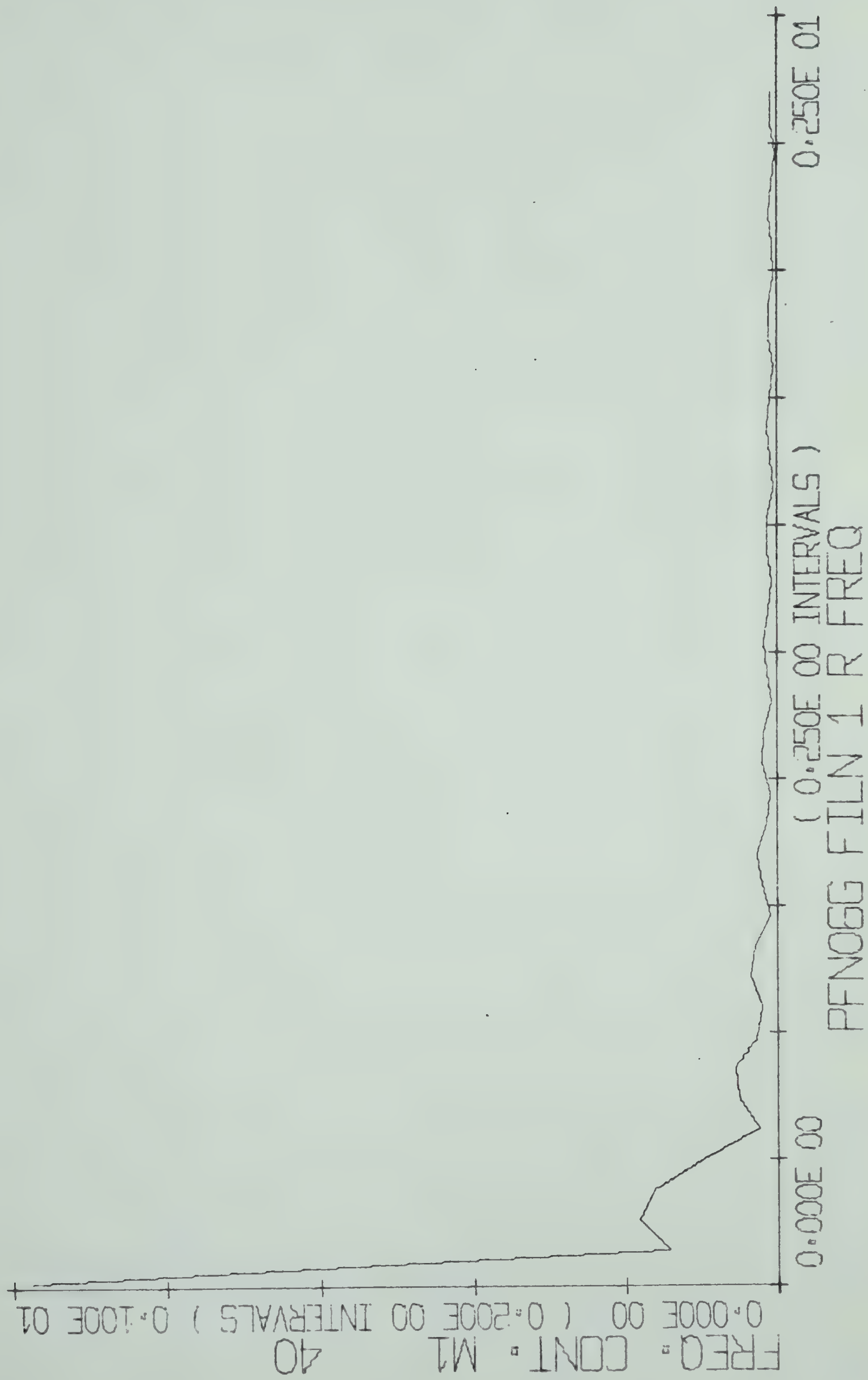

FREQ. CONT. N2 125
0.000E 00 (0.200E-01 INTERVALS) 0.120E 00
PFTOGJ FAST 1 F FREQ
0.000E 00 (0.400E 00 INTERVALS) 0.400E 01

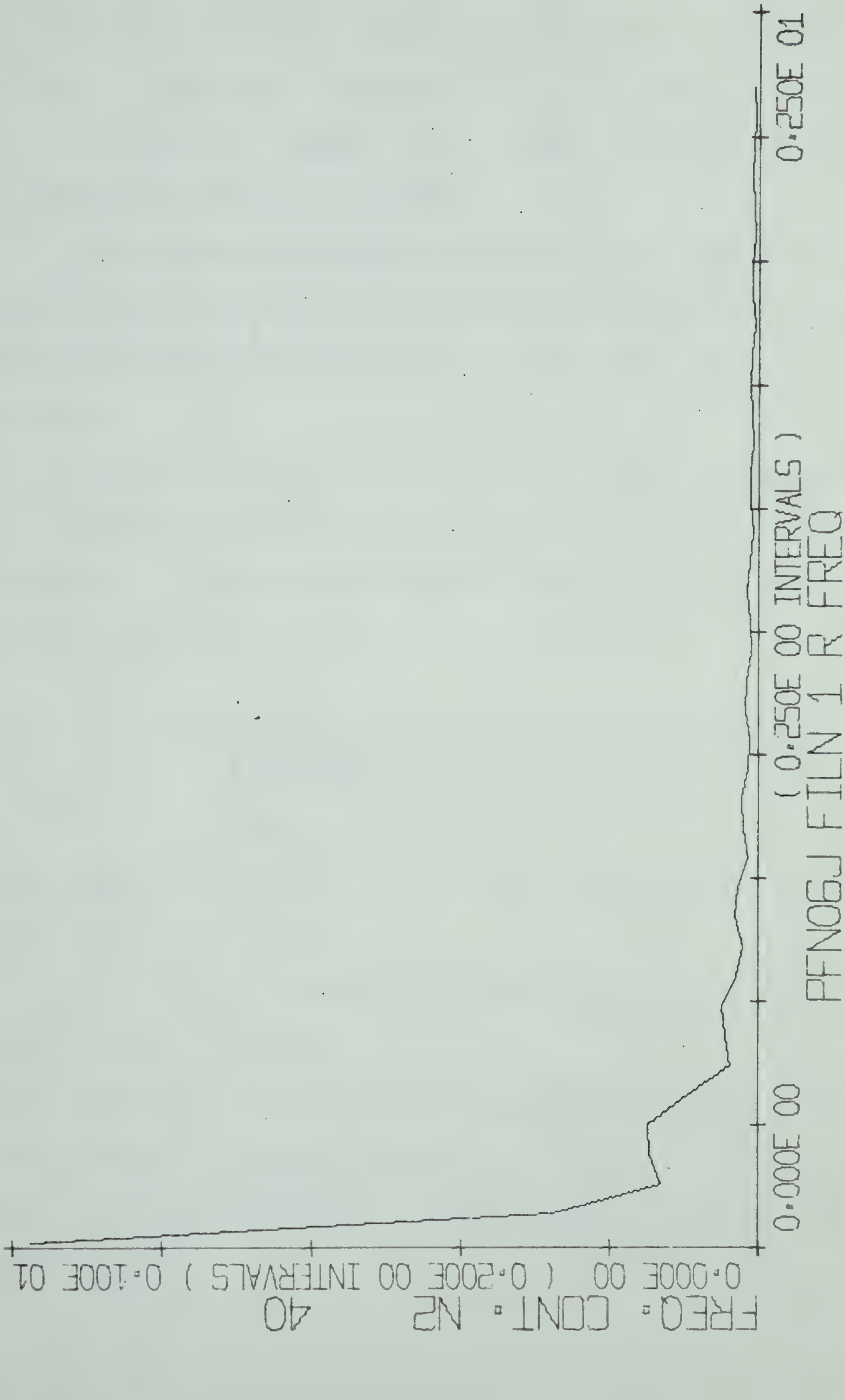

0.100E 03
0.100E-01
LOG CYCLES
PFNOGE FILN 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO M1 40


0.100E-01
0.100E 03
LOG CYCLES
PFNOGH FILN 1 R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO N2 40



0.000E 00
0.250E 01
(0.249E 00 INTERVALS)
PFNOGC FILN 1 R FREQ
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01
FREQ. CONT. R
40


0.250E 01
(0.250E 00 INTERVALS)
PFNOGG FILN 1 R FREQ
0.000E 00
FREQ. CONT. M1 40
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01


0.000E 00
(0.250E 00 INTERVALS)
0.250E 01
PFNO6J FILN 1 R FREQ
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01
FREQ. CONT. N2 40

of the input pulse. Because of this, the high-frequency asymptotes on the graphs outlined in Section 6.2.4.1 do not exhibit the correlation with the amplitude ratio discussed in Section 6.2.1.2. In fact, for Graph PFT03W, it is difficult to see any correlation at all between the high frequency asymptote and the plotted curve.

Graph PFN3WM and PFN3WN in Section 6.2.4.1, are examples of the effect of spectral smoothing in the presence of an error of closure. It appears that the smoothing caused deterioration of the data if it had any effect at all.

Because only the output pulse had an error of closure, the frequency content for the input pulse was unaffected. Therefore, Graph PFT03C in Appendix I is the frequency content curve related to the graphs outlined in Section 6.2.4.1.

6.2.4.1 Amplitude Ratio Graphs Illustrating the Effect of an Output Pulse Error of Closure With and Without Spectral Smoothing

General comments:

a. Table 4 describes the graphs presented on pages 169 - 175 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs was given in Section 6.1.

c. Graphs PFN3WM and PFN3WN do not have graph identification numbers which conform with the code outlined in Section 6.1. These two graphs illustrate the effect of spectral smoothing when an error of closure exists

d. The error of closure on an output pulse is defined as the difference between the value of the last output time point considered in the Fourier transform and the steady state value of the output before the pulse. The percent error of closure is calculated by dividing the error of closure by the maximum deviation of the output from the steady state during the pulse, i.e., the maximum change in the output during the pulse

e. Data presented in these graphs were derived from the input-output pulses presented in Graphs PTH03A and PTH03B located in Appendix I.

f. The error of closure was obtained by choosing the number of time points used in calculations so that the input pulse was closed and the output pulse was not.

g. Two Fourier transform techniques were used to derive these graphs:

   1. Fast Fourier transform (FT)

   2. Filon's quadrature (FN)

h. Some graphs have been plotted using both a continuous curve and a series of points. The graphs using points are included to provide an easy means for comparison with experimental results which are presented in point form and involve errors of closure.

i. All graphs are presented on the same scale to facilitate comparisons.

j. The sample interval on the time data used in these runs was 0.1 seconds.

k. The theoretical high-frequency asymptote for the amplitude ratio has been drawn on the plots along with a "+" at the theoretical position of the amplitude ratio curve at the corner frequency

TABLE 4

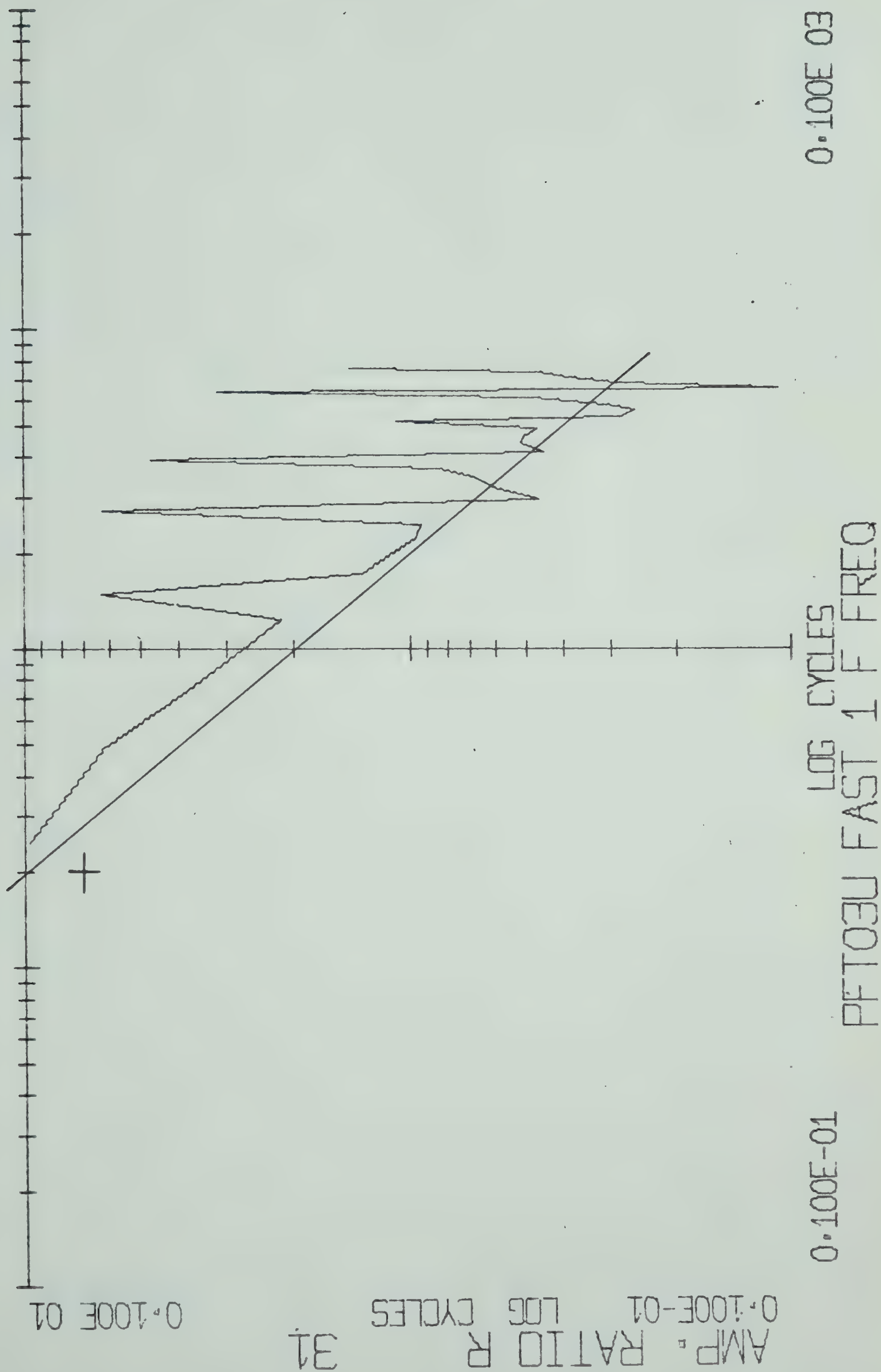
LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF AN OUTPUT PULSE ERROR OF CLOSURE

GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	PERCENT ERROR OF CLOSURE	COMMENT
PFT03U	169	FT	1.56	Amplitude ratio. This graph was obtained from the same pulse data used for Graph PFT03A (Section 6.2.1.3) but a truncation error of closure was introduced. (A change in number of time points processed makes this plot difficult to compare to PFT03A because the FT frequency range is a function of the number of time points; see Section 3.2.4.
PFT03U	170	FT	1.56	Point graph using the data plotted in the preceding graph.
PFN03U	171	FN	0.96	Amplitude ratio. This graph was obtained from the same pulse data used for Graph PFN03A, (Section 6.2.1.3) but a truncation error of closure was introduced.

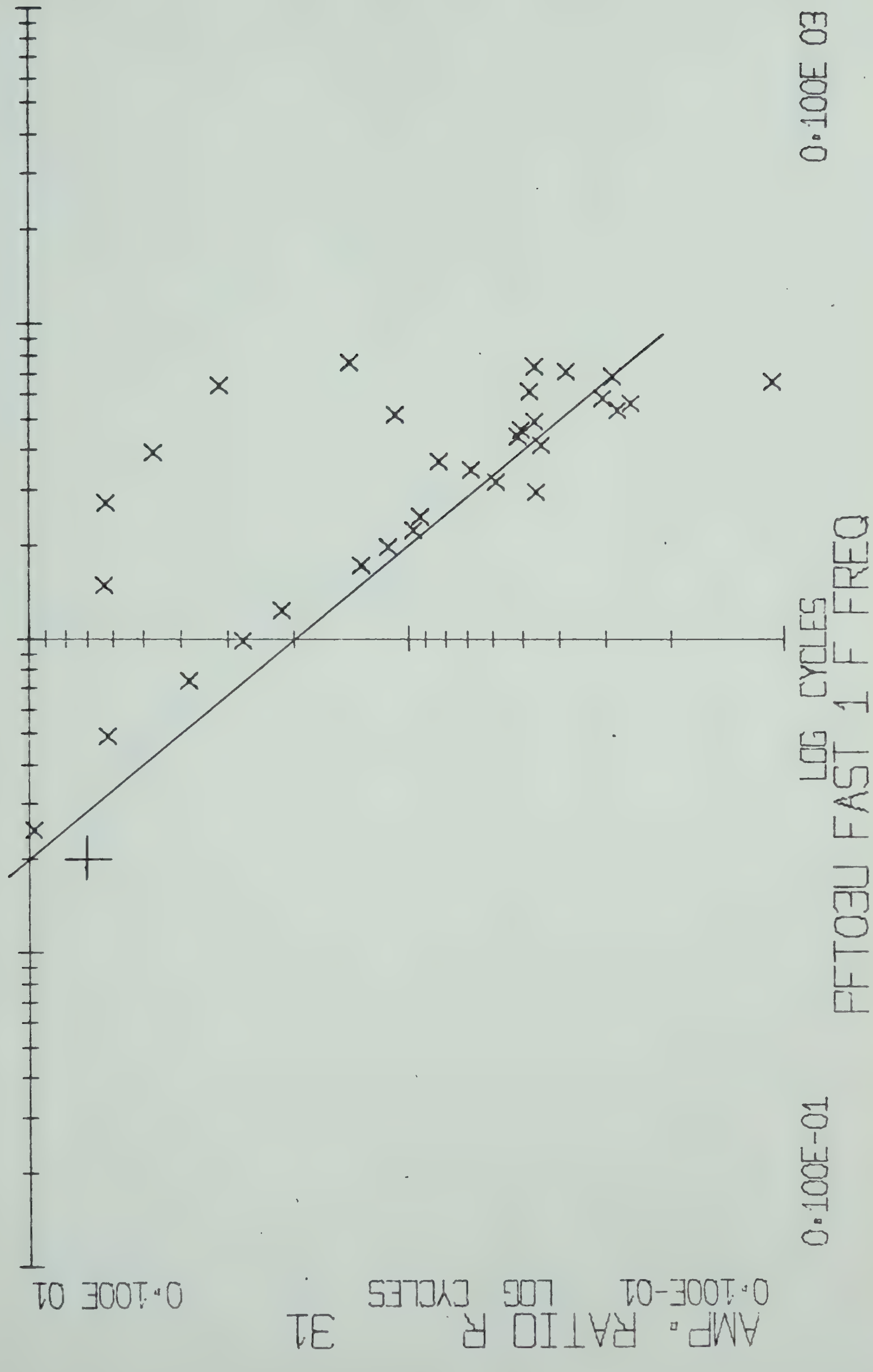
TABLE 4 (CONT'D)

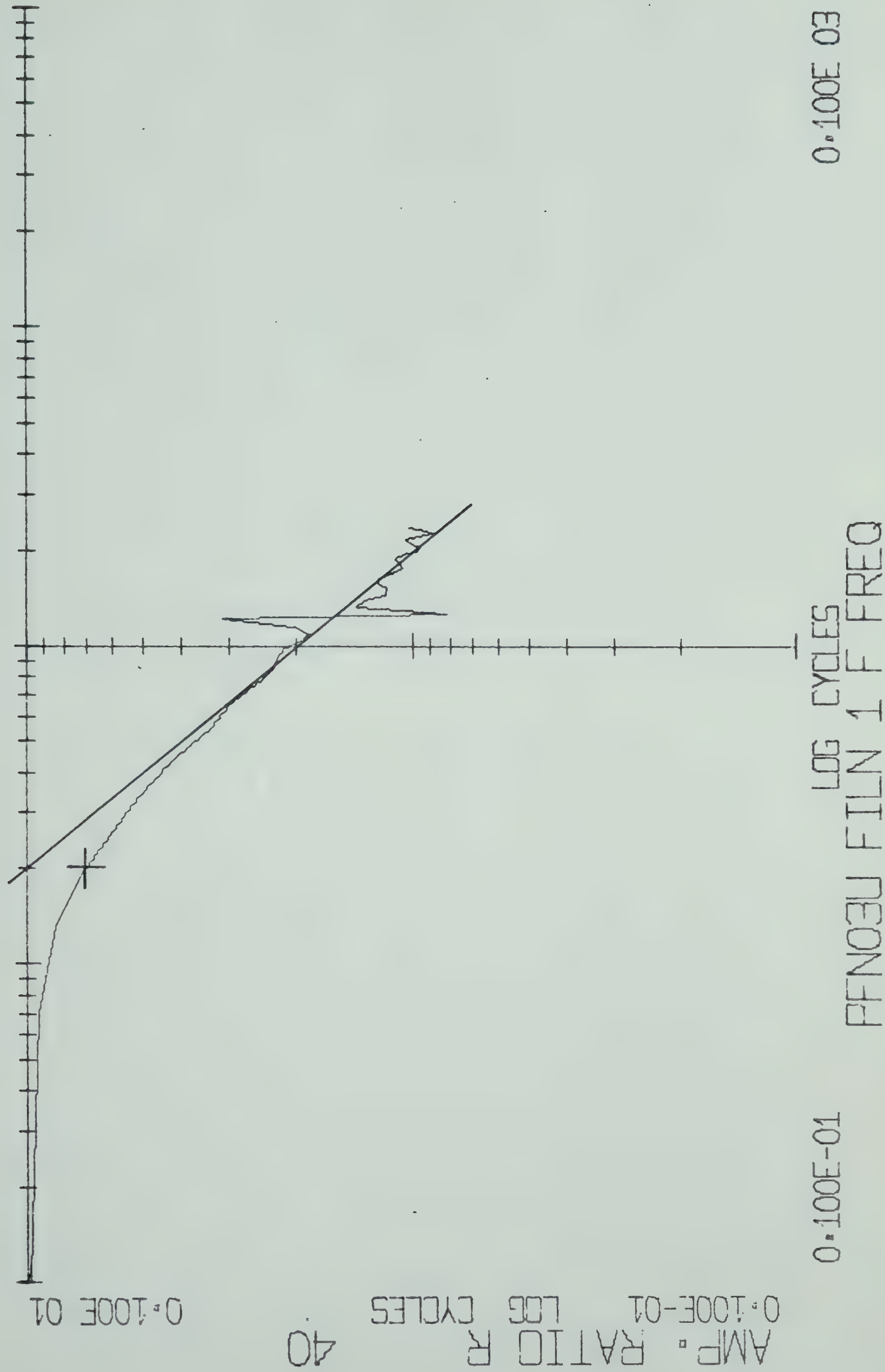
LIST OF FREQUENCY RESPONSE GRAPHS ILLUSTRATING THE EFFECT OF AN OUTPUT PULSE ERROR OF CLOSURE

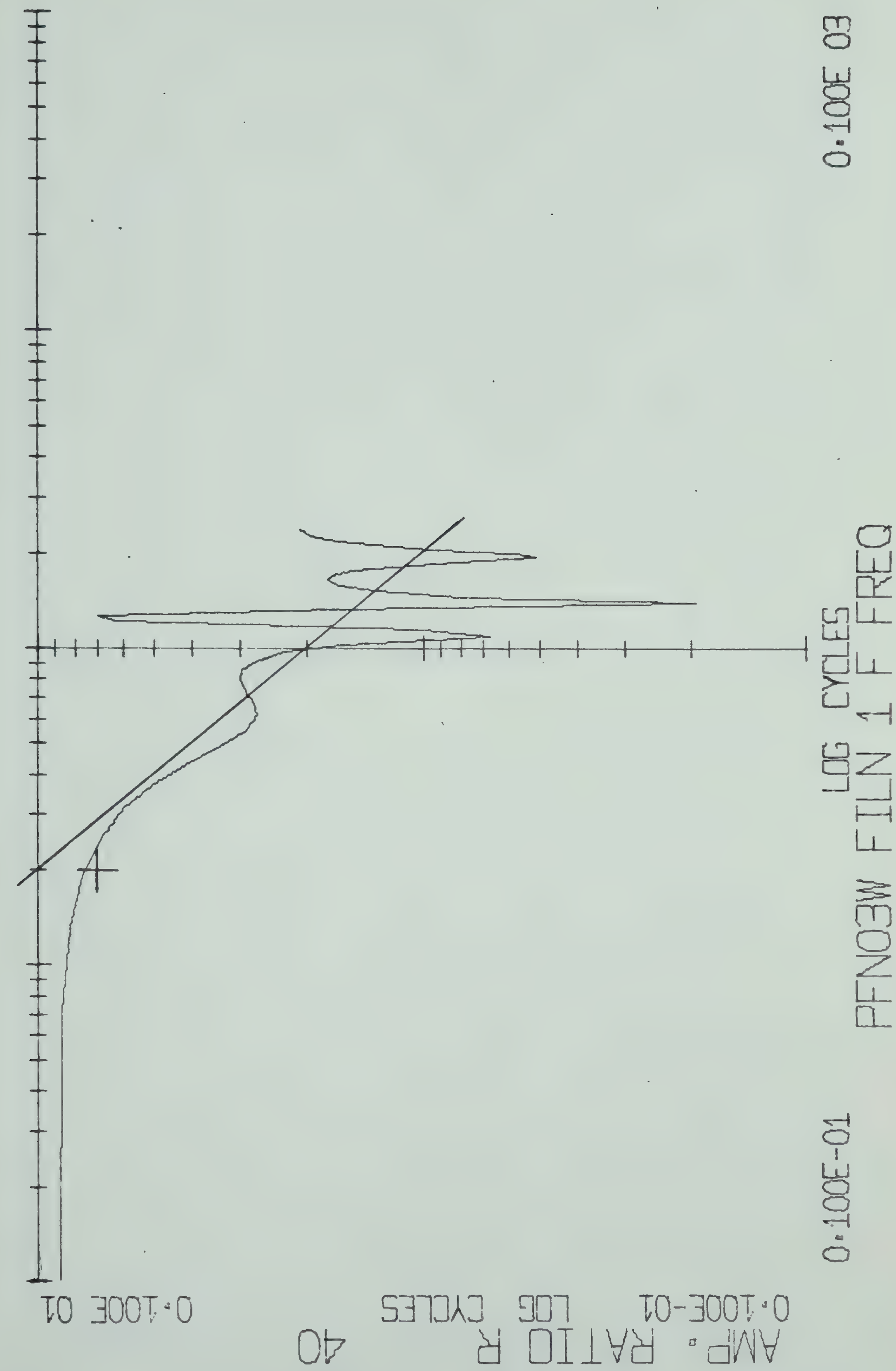
GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	PERCENT ERROR OF CLOSURE	COMMENT
PFN03W	172	FN	0.96	Amplitude ratio. This graph was obtained from the same pulse data used for Graph PFN03A (Section 6.2.1.3) but a truncation error of closure was introduced.
PFN03W	173	FN	10.7	Point graph using the data plotted in the preceding graph.
PFN3WM	174	FN	10.7	Amplitude ratio. The same data as used for Graph PFN03W but hamming smoothing has been applied.
PFN3WN	175	FN	10.7	Amplitude ratio. The same data as used for Graph PFN03W but hanning smoothing has been applied.

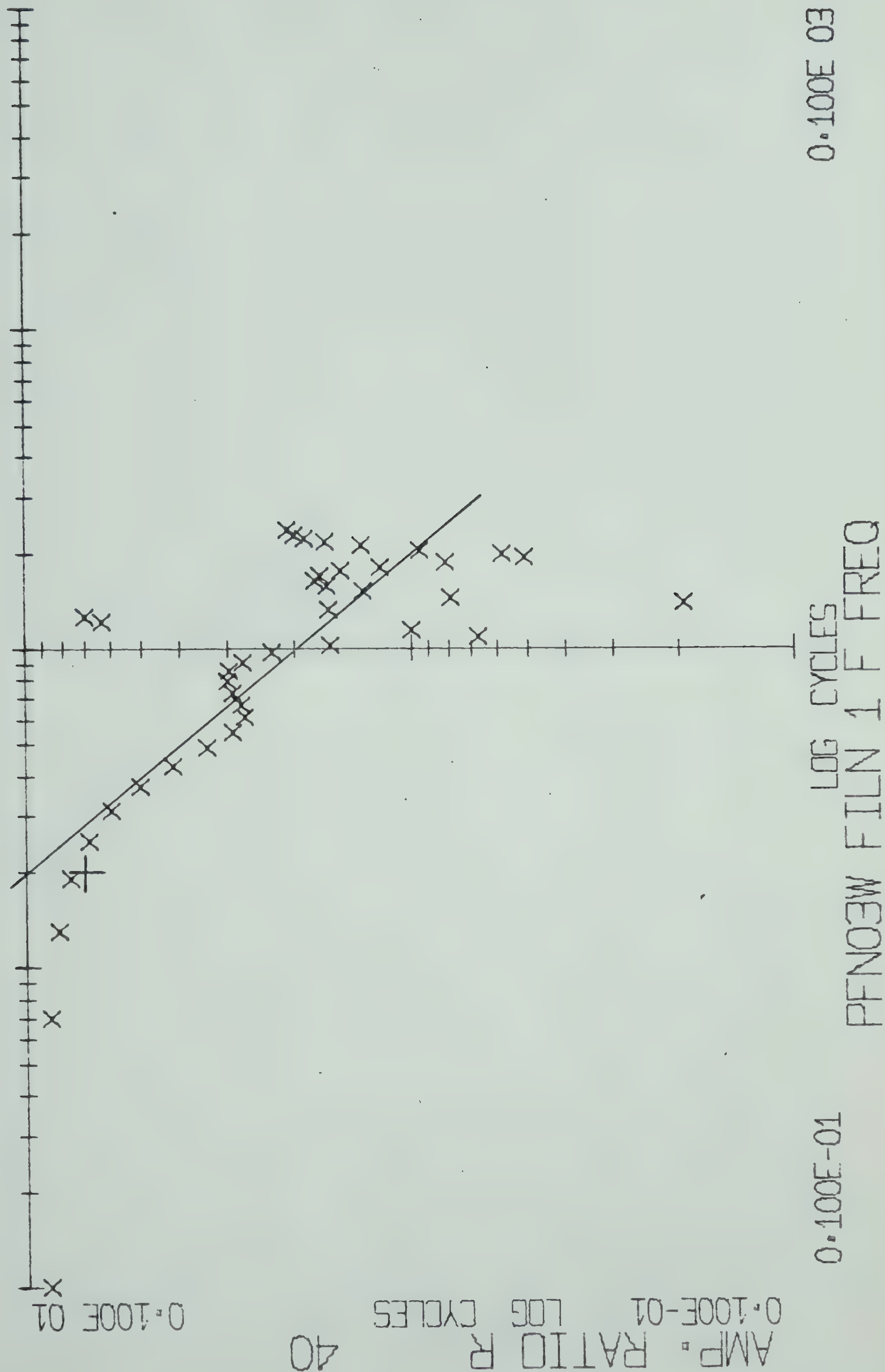


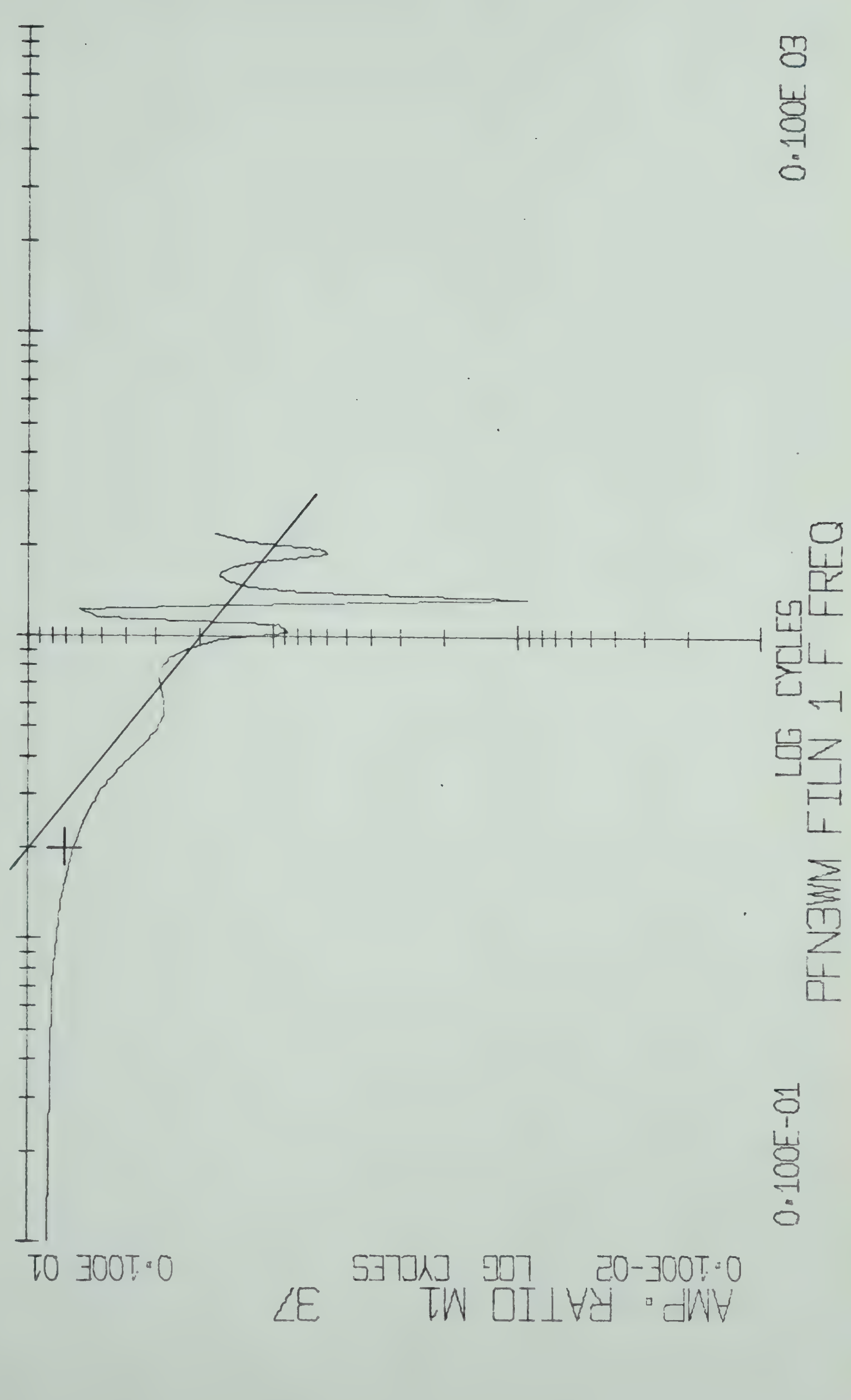
0.100E 03
LOG CYCLES
PFTO3U FAST 1 F FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
31

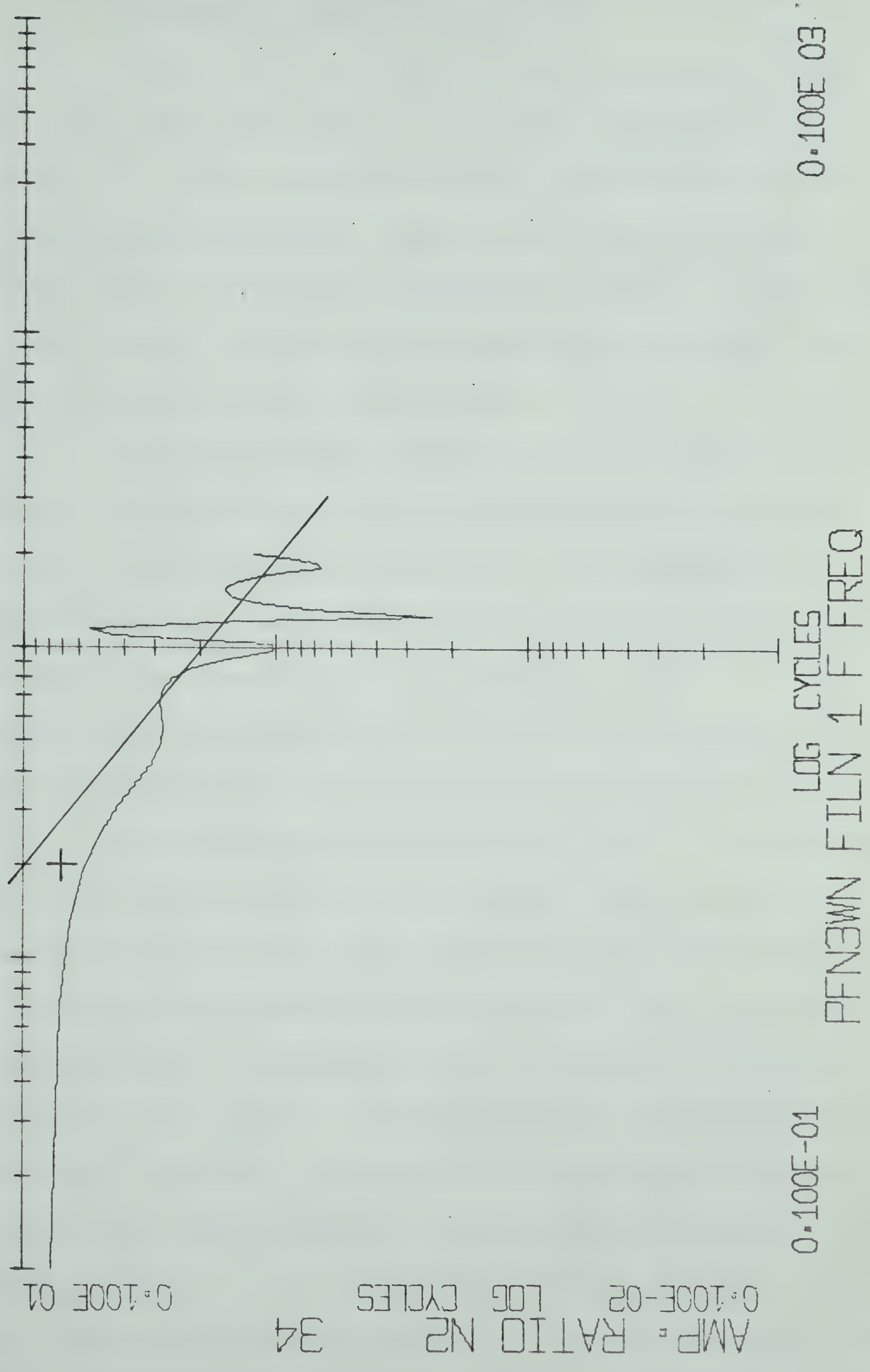

0.100E 03
0.100E-01
LOG CYCLES
PFTO3U FAST 1 F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
31


AMP. RATIO R
40
0.100E-01 LOG CYCLES 0.100E 01
LOG CYCLES
PFNO3U FILN 1 F FREQ
0.100E-01
0.100E 03


0.100E 03
LOG CYCLES
PFNO3W FILN 1 F FREQ
0.100E-01
0.100E 01
LOG CYCLES
0.100E-01
AMP. RATIO R
40


0.100E 03
LOG CYCLES
PFNO3W FILN 1 F FREQ
0.100E-01
AMP. RATIO R 40
0.100E 01
LOG CYCLES
0.100E-01


0.100E 03
0.100E-01
LOG CYCLES
PFN3WM FILN 1 F FREQ
0.100E 01
LOG CYCLES
0.100E-02
AMP. RATIO M1 37


0.100E 03
LOG CYCLES
PFN3WN FILN 1 F FREQ
0.100E-01
0.100E-02 LOG CYCLES 0.100E 01
AMP. RATIO N2 34

6.2.5 Two Cascaded First Order Systems and a Second Order Underdamped System

Theoretical runs 07 and 08 involved pulse testing two cascaded first order systems and a second order underdamped system, respectively. The object of these runs was to demonstrate that the pulse analysis procedures used in this work were valid for systems other than a simple first order. The parameters used for the two systems tested were arbitrarily chosen and are given in Section 6.2.5.1 along with all the graphs associated with each run.

The amplitude ratio and phase lag graphs obtained are characteristic of the systems tested except for an unnatural attenuation at the low frequency end of the curves. This attenuation is attributed to the large sample interval and narrow input pulse which was discussed in Section 6.2.2 for the first order system. It is interesting to compare the effect of pulse durations and sample interval (items f and g, Section 6.2.5.1) on the amount of low frequency attenuation in the amplitude ratio graphs in Section 6.2.5.1. In Graph PFT07A, obtained using a sample interval of 0.2 seconds, there is less low frequency attentuation than in Graph PFN07A, which was obtained with a sample interval of 0.6 seconds. In both cases the input pulse duration was 2 seconds and the time constant of the two cascaded first order systems were 2 and 10 seconds. This behaviour is in agreement with that discussed in Section 6.2.2 where the low frequency attentuation for a first order system was greater when the ratio of sample interval to input pulse duration was high. On the other hand, for run 08 the low frequency attentuation in Graph PFT08A and in Graph PFN08A is the reverse,

or at least does not conform with, the previous behaviour. Therefore, the results in Section 6.5.2.1 further demonstrate the need, expressed in Section 6.2.2, for additional study into the effect the time sample interval has on the position of the amplitude ratio curve.

6.2.5 1 Time and Frequency Domain Graphs Derived by Pulse Testing Two Cascaded First Order Systems and a Second Order System

General comments:

a. Table 5 describes the graphs presented on pages 182 - 195 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs was given in Section 6.1.

c. All of the graphs associated with runs 07 and 08 are presented on pages 182 - 195. The input and output pulse data was obtained using the methods outlined in Chapter IV.

d Three Fourier transform techniques were used to derive the amplitude ratio and phase lag graphs:

1. Fast Fourier transform (FT)

2. Trapezoidal quadrature (TP)

3. Filon's quadrature (FN)

e. Similar graphs are not necessarily on the same scale so careful consideration should be given to scales when comparisons are made.

f The duration of the input pulses applied to the two systems were:

1. cascaded first order systems - 2 seconds

2. second order - 1 second

g. The time data sample intervals were as follows:

1. Using the FT technique:

(i) cascaded first order systems - 0.2 seconds

(ii) second order - 0.2 seconds

2. Using the TP and FN techniques:

(i) cascaded first order systems - 0.6 seconds

(ii) second order - 0.4 seconds

h. The cascaded first order systems had time constants of 2 and 10 seconds. The second order had a natural frequency of 0.5 radians per second and a damping ratio of 0.1.

i. The theoretical high-frequency asymptote for the amplitude ratio has been drawn on the graphs along with a "+" at the theoretical position of the amplitude ratio at the corner frequency of the first order systems.

j. For the cascaded first order systems, the straight line approximation to the phase lag for each first order system appears as a dotted line. A "+" appears at the corner frequency of each first order system. At the corner frequency of the second first order system a third "+" appears, which is a graphic summation of the individual phase lags at that frequency. For the second order system phase lag a "+" appears at the theoretical location of the phase lag at the natural frequency of the system.

k. The general arrangement of the graphs presented in this section is as follows:

1. input pulse

2. frequency content of input pulse

3. output pulse

4. amplitude ratio derived by FT

5. phase lag derived by FT

6. amplitude ratio derived by TP or FN

7. phase lag derived by TP or FN

6.3 Experimental Analysis

In the experimental analysis time domain pulse data were obtained from the test heat exchanger process using the technique outlined in Chapter V. In these experimental tests, rectangular and ramp-shaped tube flow (cold fluid) pulses were used as input pulses. The output pulse for all runs was the outlet temperature of the counter-flowing shell stream (hot fluid). For each run the steady state conditions before the pulse were approximately:

a. Tube flow - 8.00 gpm (U.S.)

b. Shell flow - 15.00 gpm (U.S.)

c. Tube temp - in = 57.9°F; out = 80.8

d. Shell Temp - in = 149.5°F; out = 137.1

The maximum amplitude for the flow pulse was approximately 12.00 gpm (U.S.) for every run meaning a magnitude change of fifty percent of the steady state.

The experimental run numbers range from 10 to 20 inclusive with number 15 being deleted owing to loss of some of the data. It should be noted that graphs derived from experimental runs are not arranged in order of run number but rather in order of input pulse duration starting with the rectangular shape and following with the ramp shape.

TABLE 5

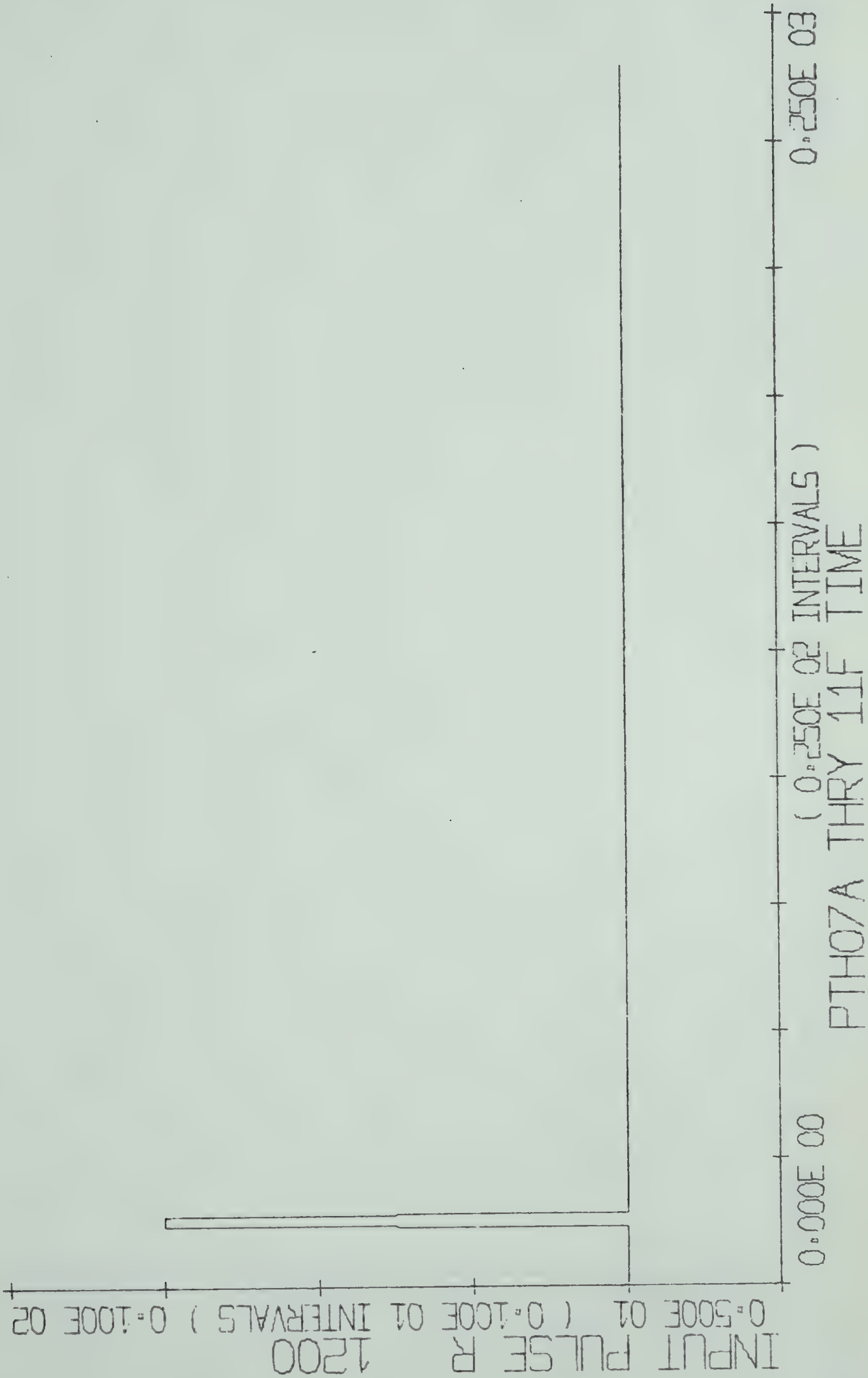
LIST OF GRAPHS ASSOCIATED WITH PULSE TESTING TWO
A CASCADED FIRST ORDER SYSTEMS AND SECOND ORDER SYSTEM

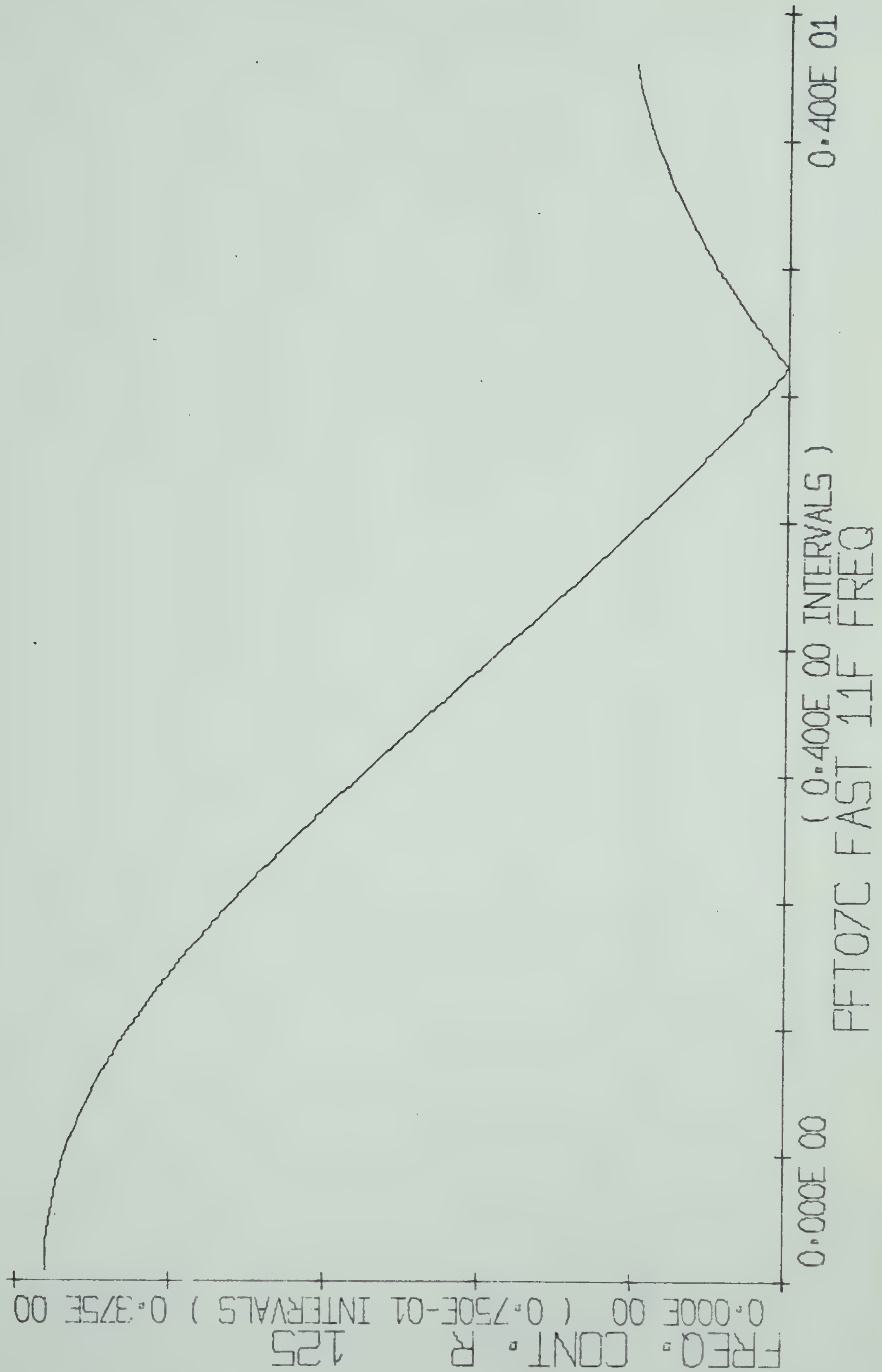
GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PTH07A	182		Input pulse applied to the cascaded first order systems.
PFT07C	183	FT	Frequency content curve of input pulse.
PTH07B	184		Output pulse from the cascaded first order systems when forced with the pulse shown in Graph PTH07A
PFT07A	185	FT	Amplitude ratio of the cascaded first order systems.
PFT07B	186	FT	Phase lag of the cascaded first order systems.
PTP07A	187	TP	Amplitude ratio of the cascaded first order systems.
PTP07B	188	TP	Phase lag of the cascaded first order systems.
PTH08A	189		Input pulse applied to the second order system.
PFT08C	190	FT	Frequency content of the input pulse.

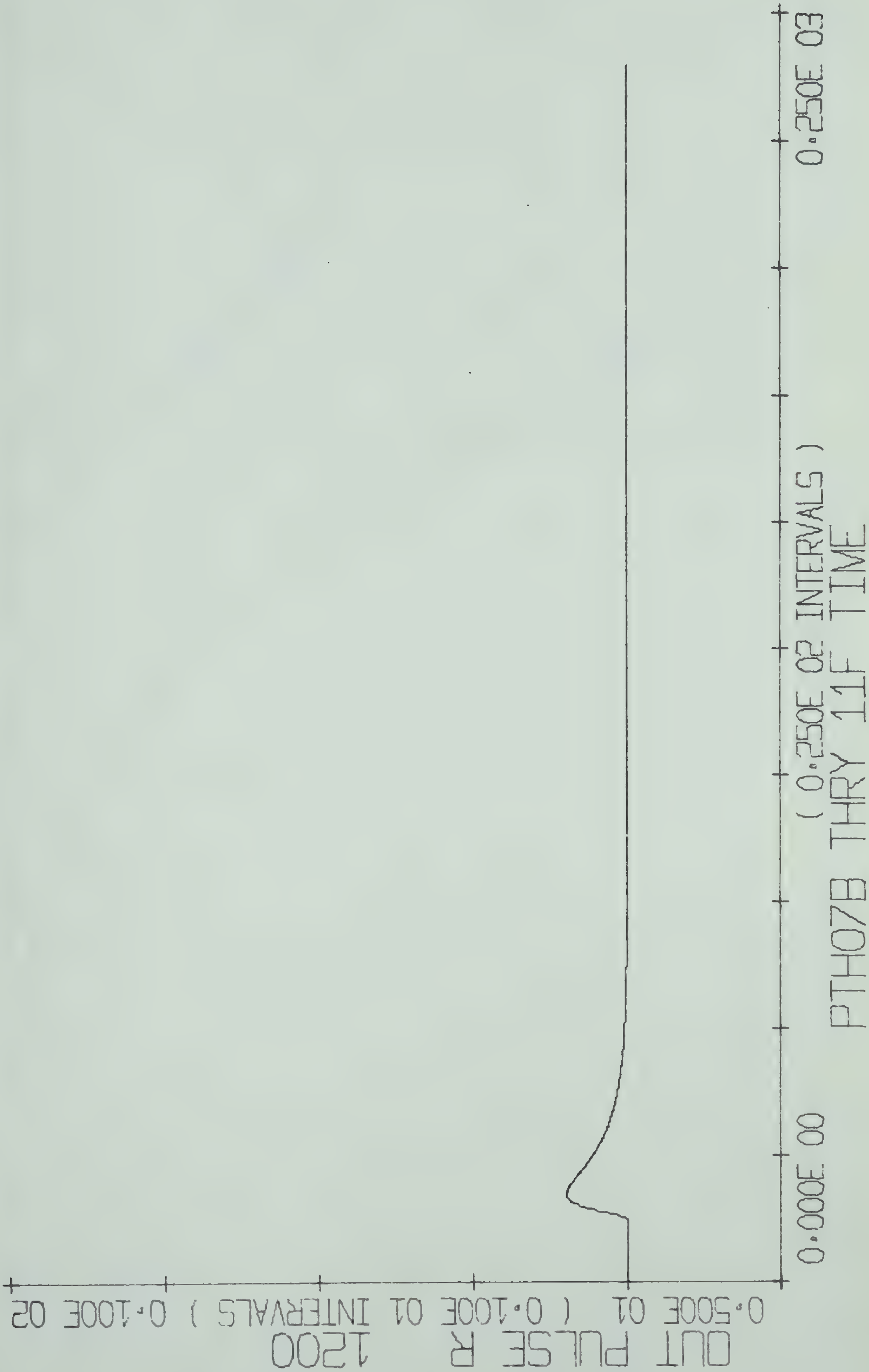
TABLE 5 (CONT'D)

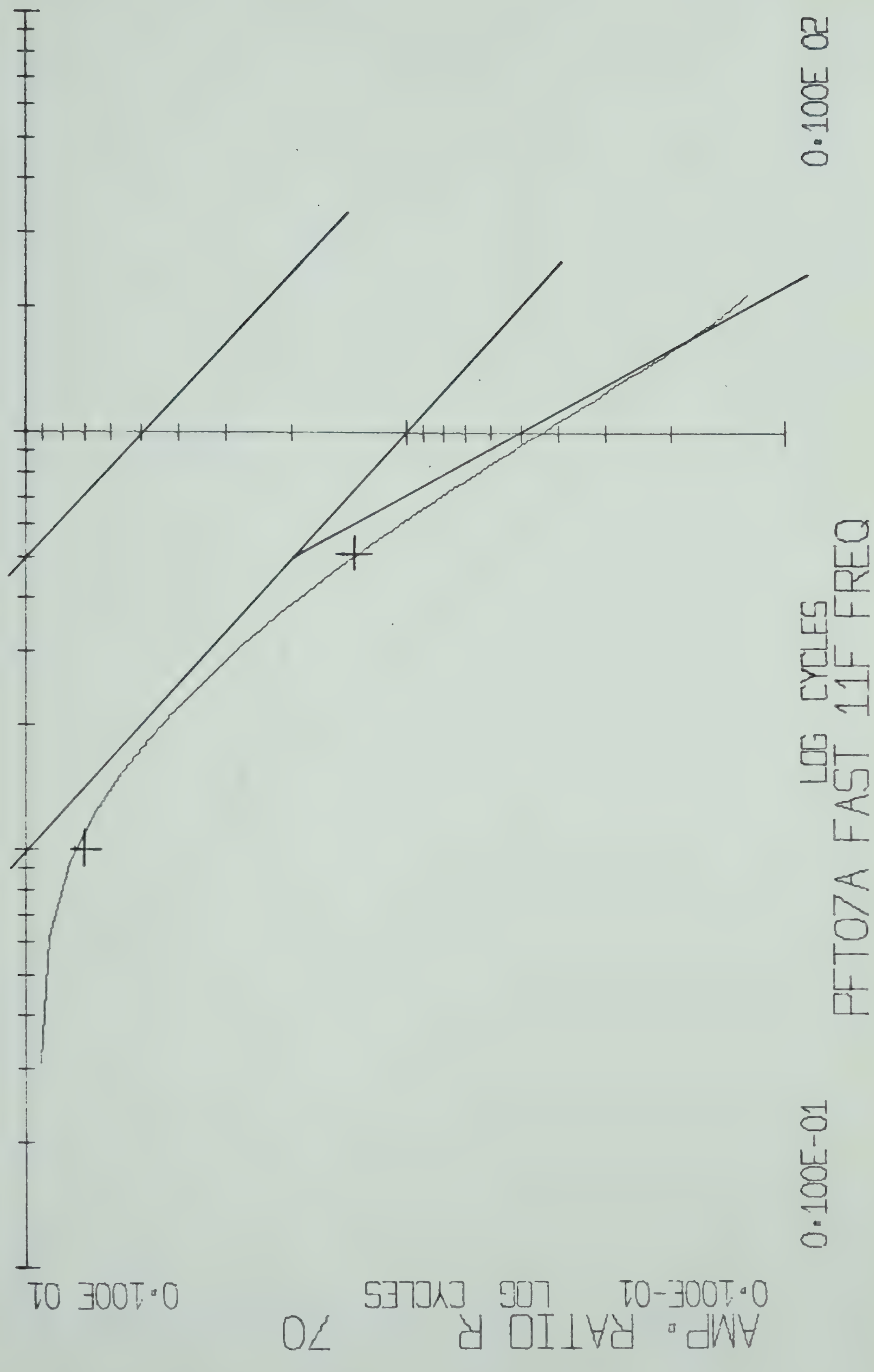
LIST OF GRAPHS ASSOCIATED WITH PULSE TESTING TWO A CASCADED FIRST ORDER SYSTEMS AND SECOND ORDER SYSTEM

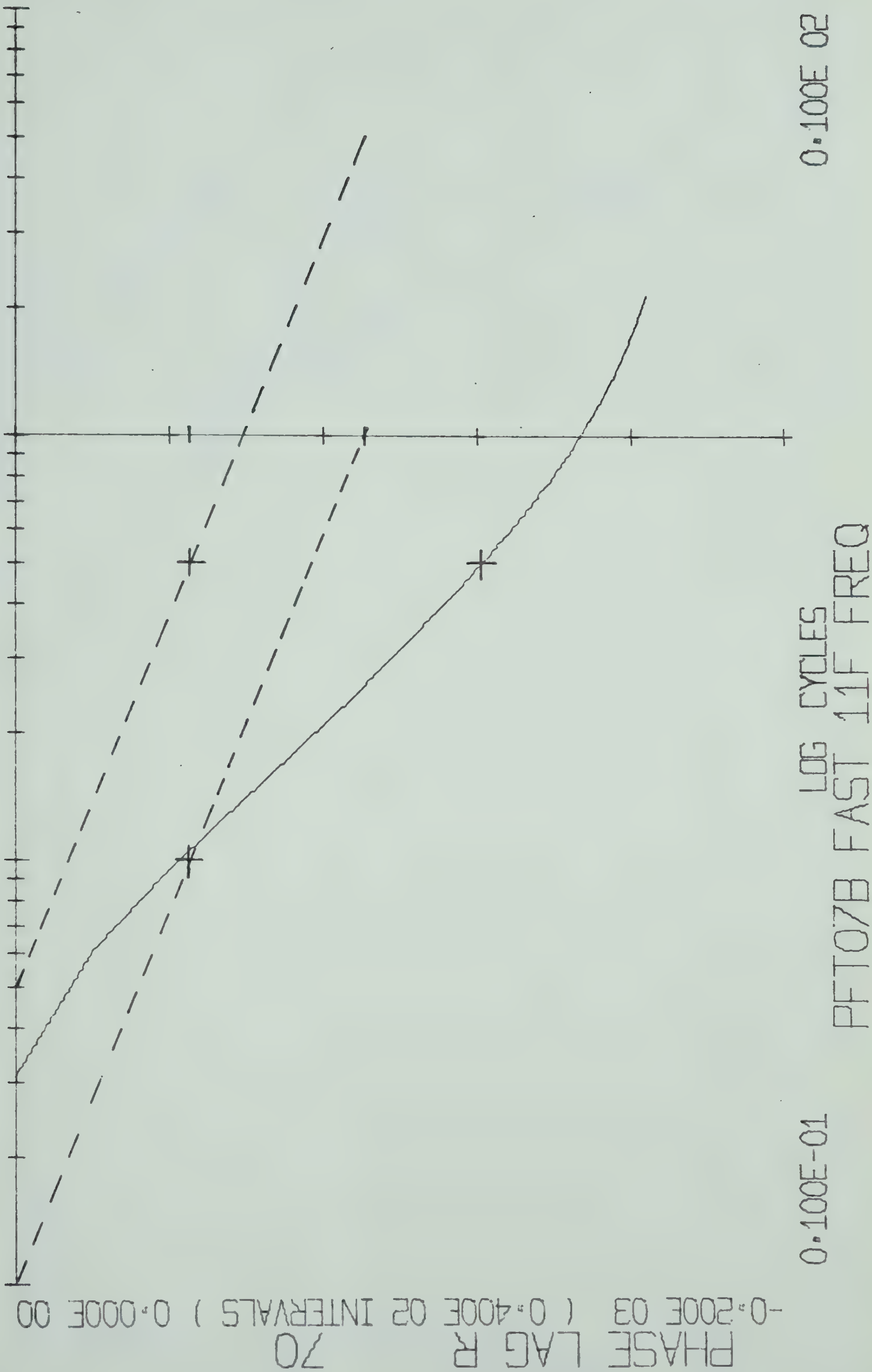
GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PTH08B	191		Output pulse from the second order system when forced with the pulse shown in Graph PTH08A.
PFT08A	192	FT	Amplitude ratio of the second order system.
PFT08B	193	FT	Phase lag of the second order system.
PFN08A	194	FN	Amplitude ratio of the second order system.
PFN08B	195	FN	Phase lag of the second order system.

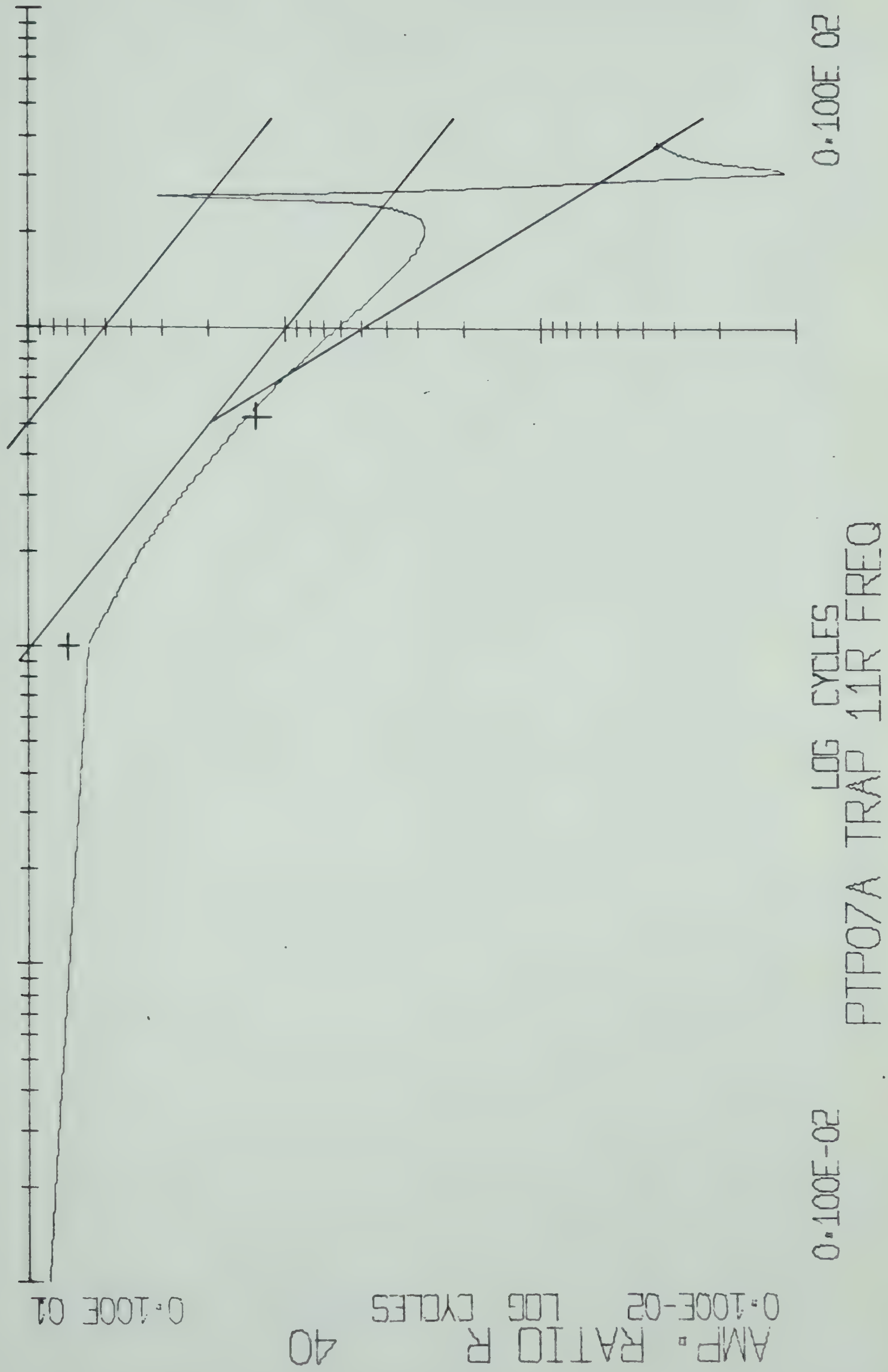

0.000E 00
(0.250E 02 INTERVALS)
0.250E 03
PTH07A THRY 11F TIME
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
INPUT PULSE R 1200

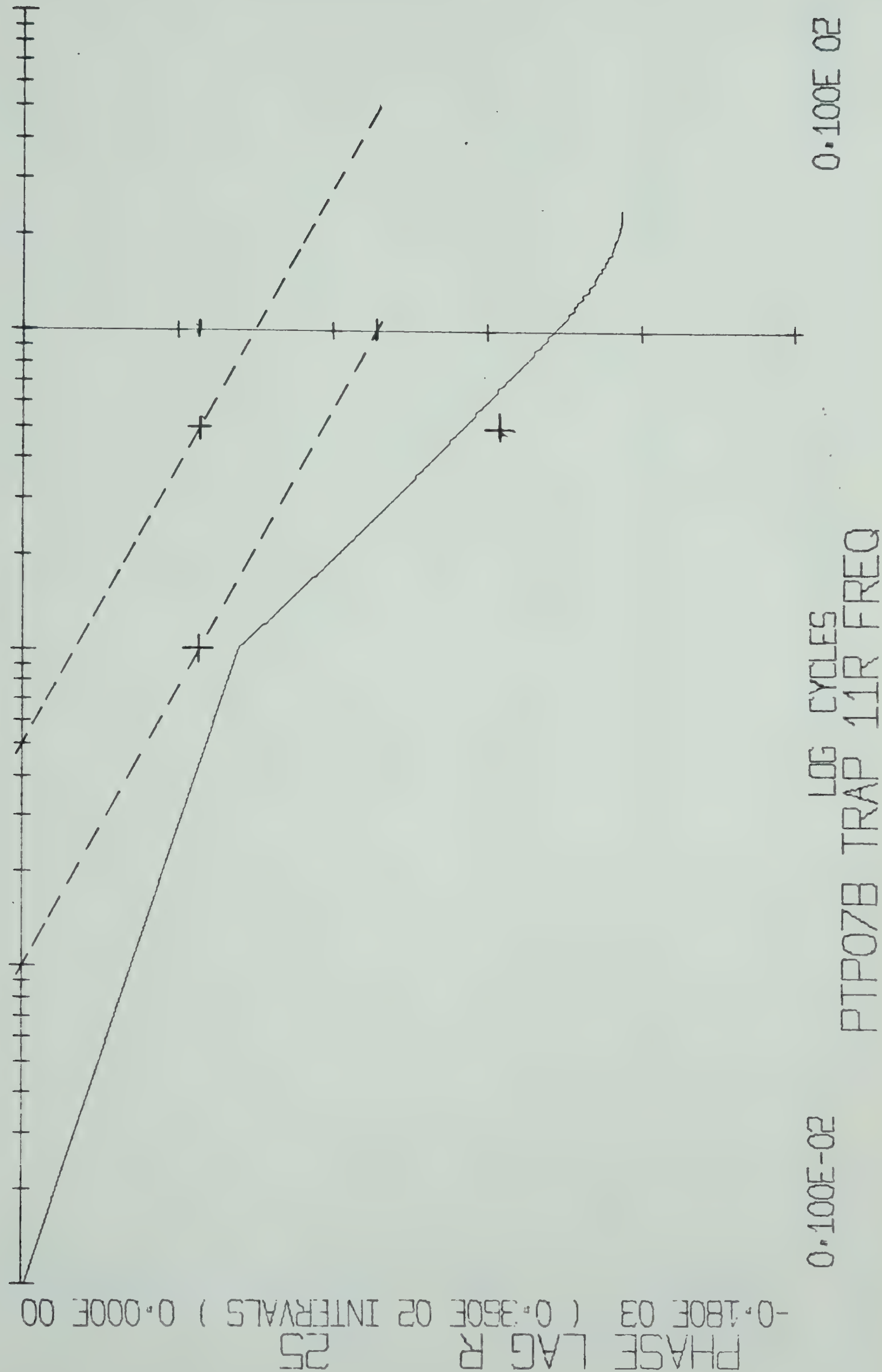

0.000E 00
0.400E 01
(0.400E 00 INTERVALS)
PFT07C FAST 11F FREQ
0.000E 00 (0.750E-01 INTERVALS) 0.375E 00
FREQ. CONT. R 125


0.000E 00
(0.250E 02 INTERVALS)
0.250E 03
PTH07B THRY 11F TIME
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
OUT PULSE R 1200

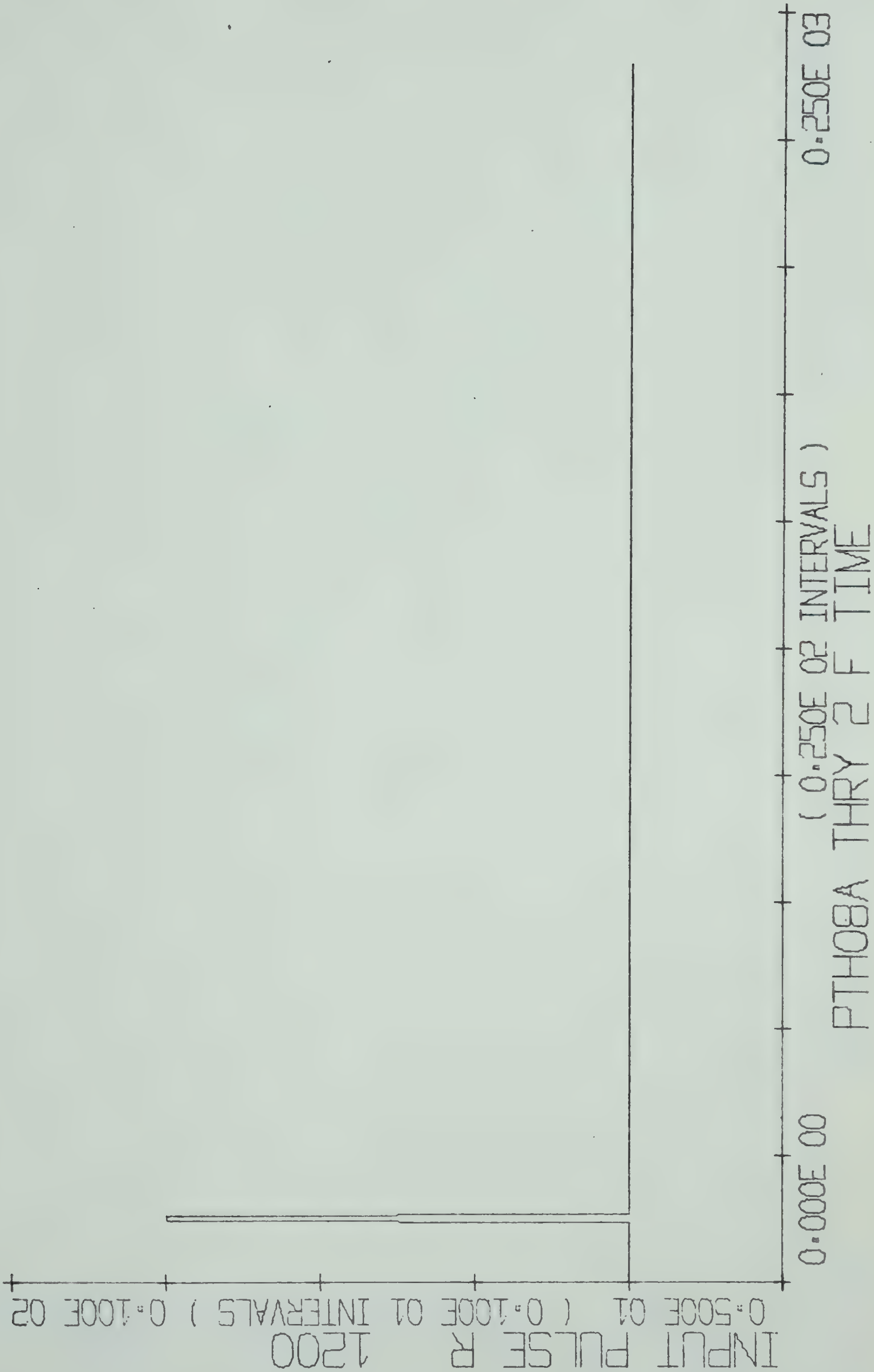

0.100E 02
0.100E-01
LOG CYCLES
PFT07A FAST 11F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
70


0.100E-01
0.100E 02
LOG CYCLES
PFT07B FAST 11F FREQ
-0.200E 03 (0.400E 02 INTERVALS) 0.000E 00
PHASE LAG R
70

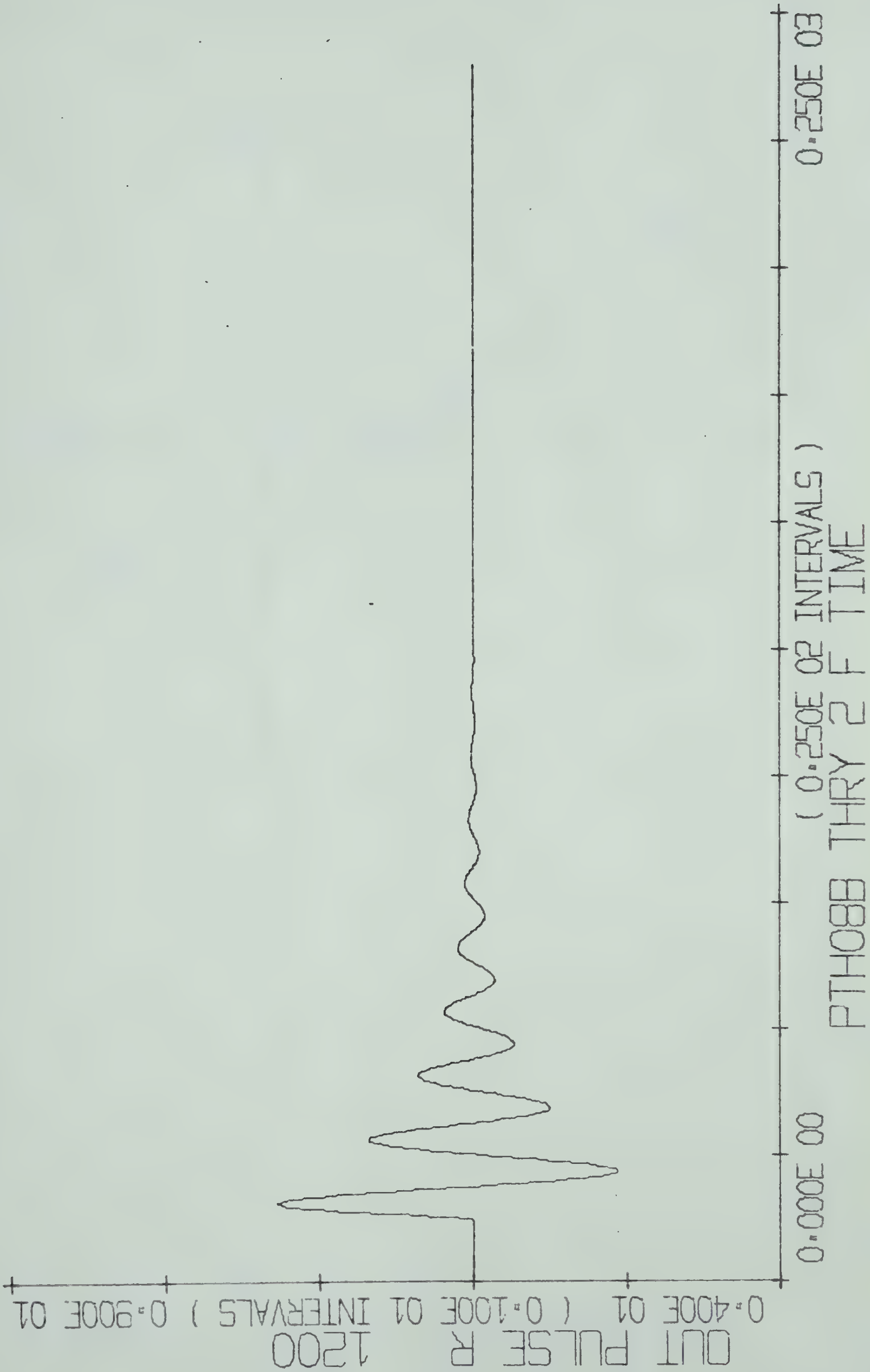

0.100E 02
0.100E-02
LOG CYCLES
PTP07A TRAP 11R FREQ
0.100E 01
0.100E-02
LOG CYCLES
AMP. RATIO R
40

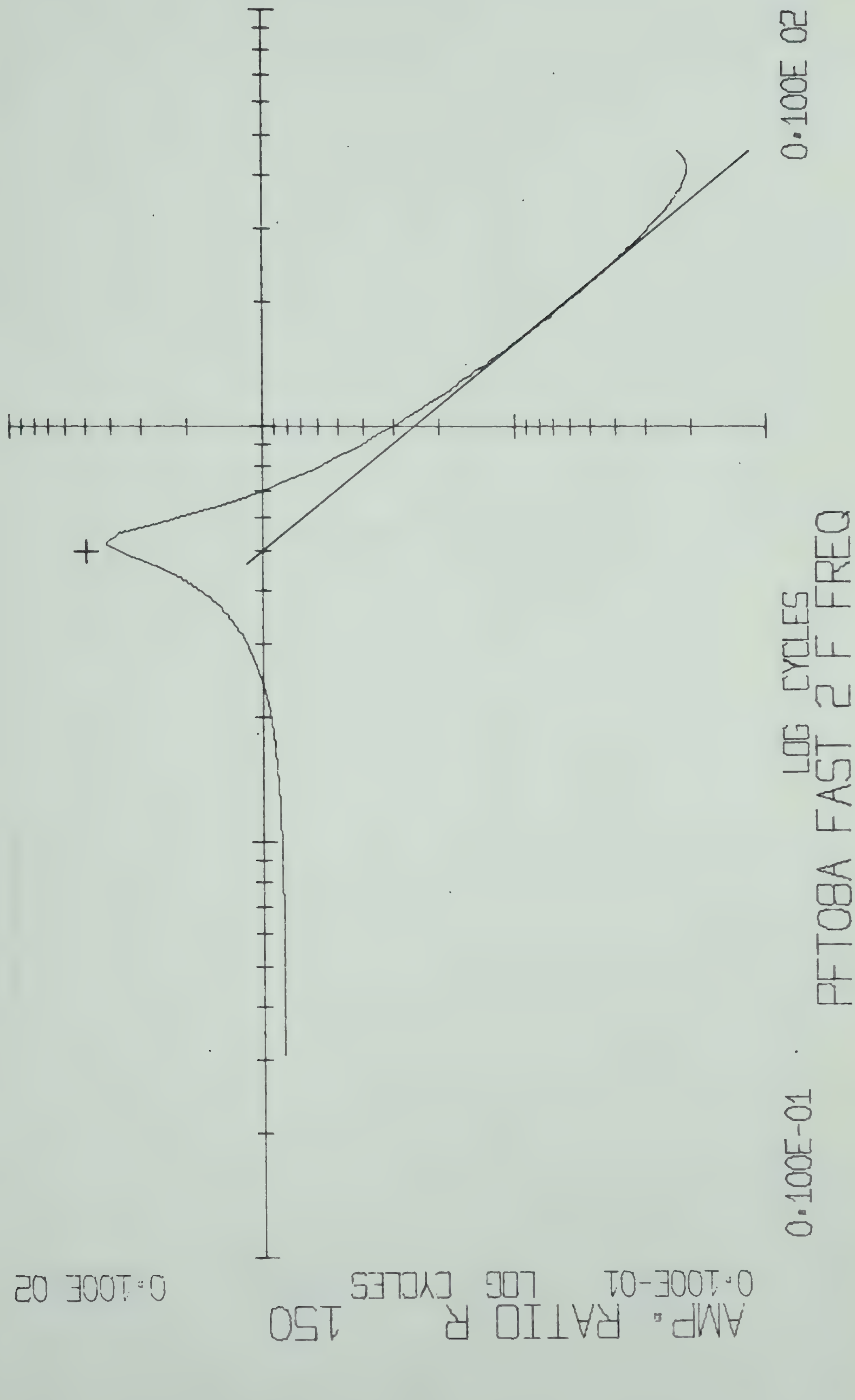


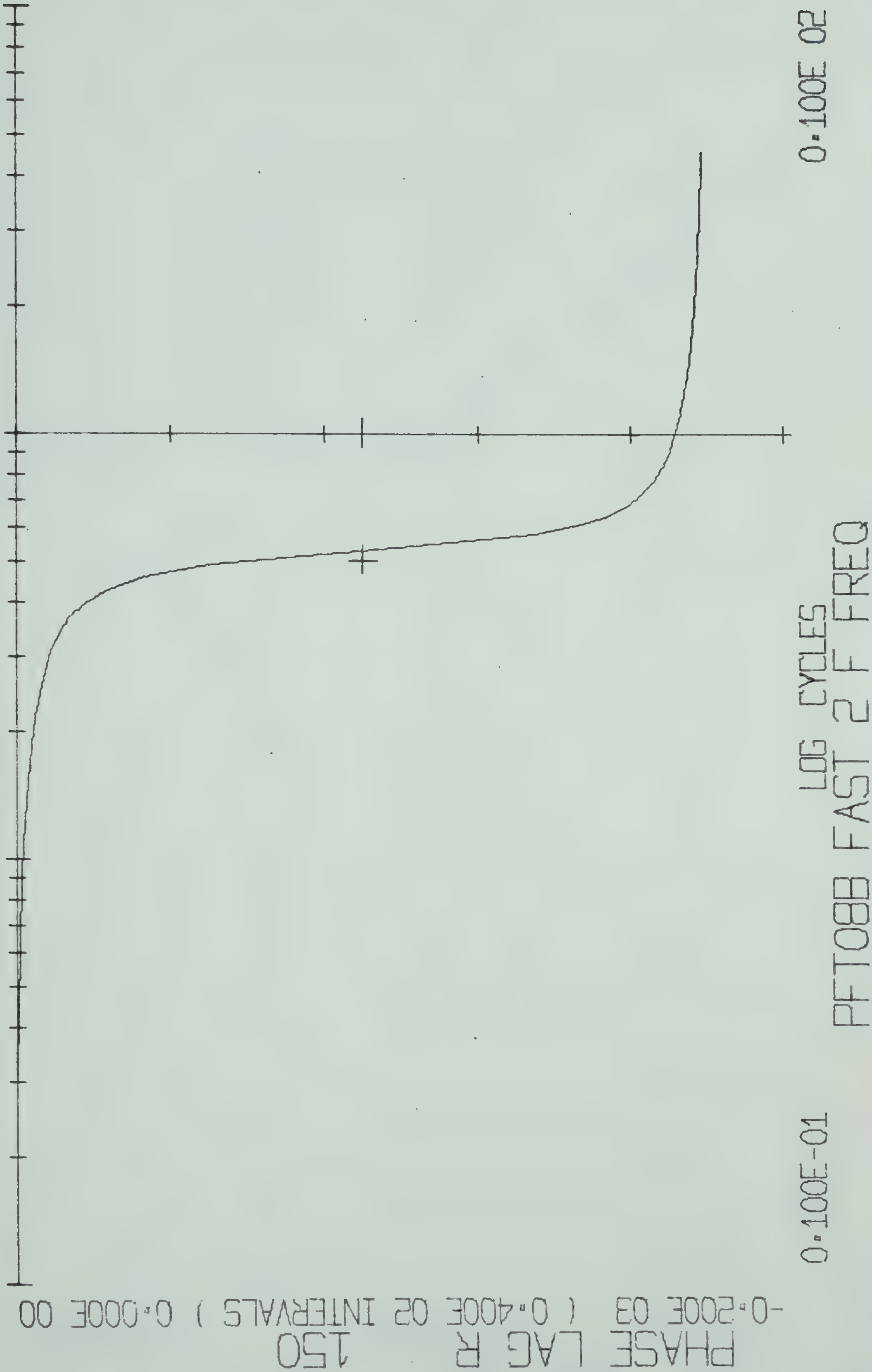
PHASE LAG R
25
-0.180E 03 (0.360E 02 INTERVALS) 0.000E 00
0.100E-02
0.100E 02
LOG CYCLES
PTP07B TRAP 11R FREQ

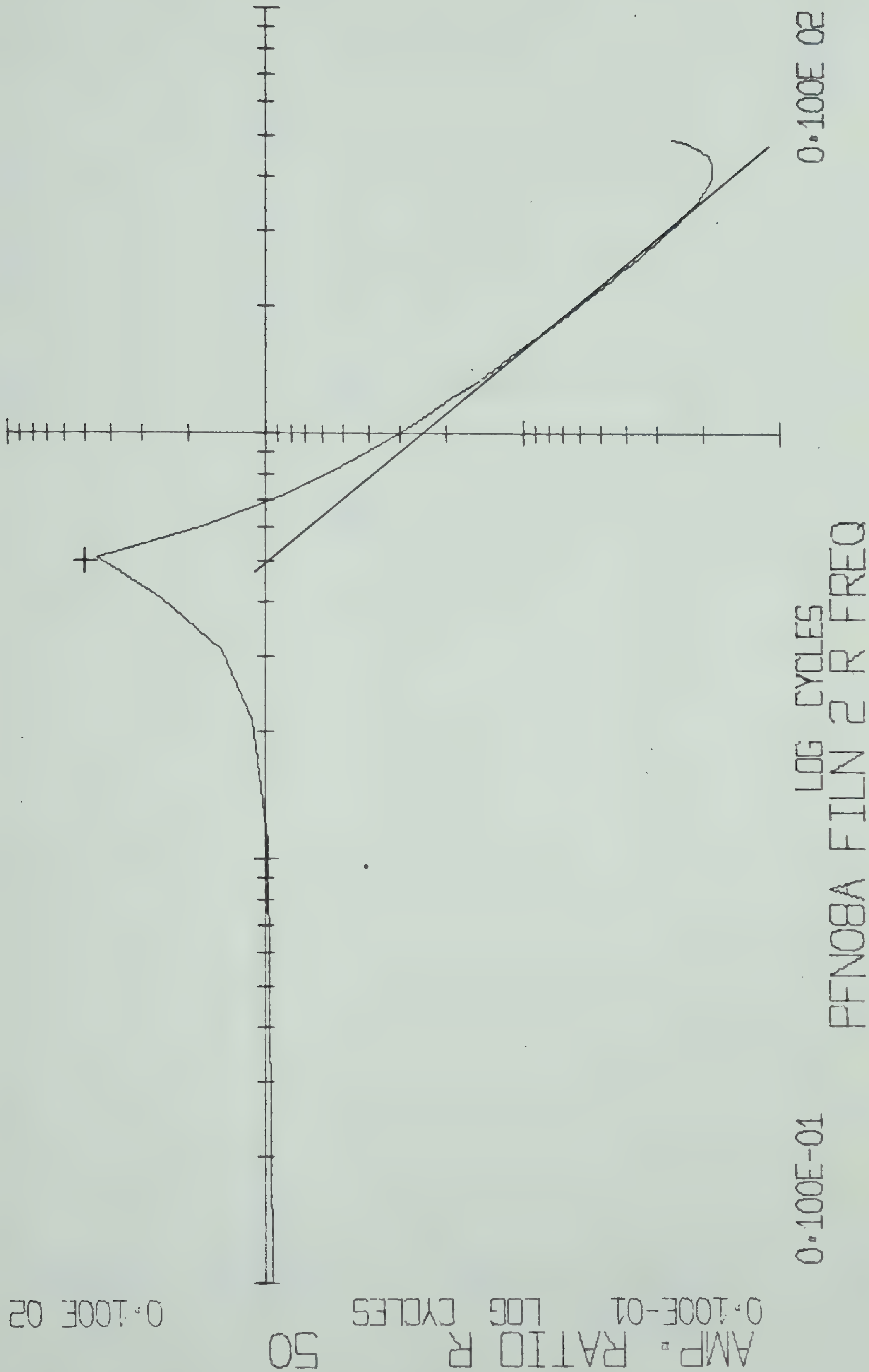

0.250E 03
(0.250E 02 INTERVALS)
PTHOBA THRY 2 F TIME
0.000E 00
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
INPUT PULSE R 1200

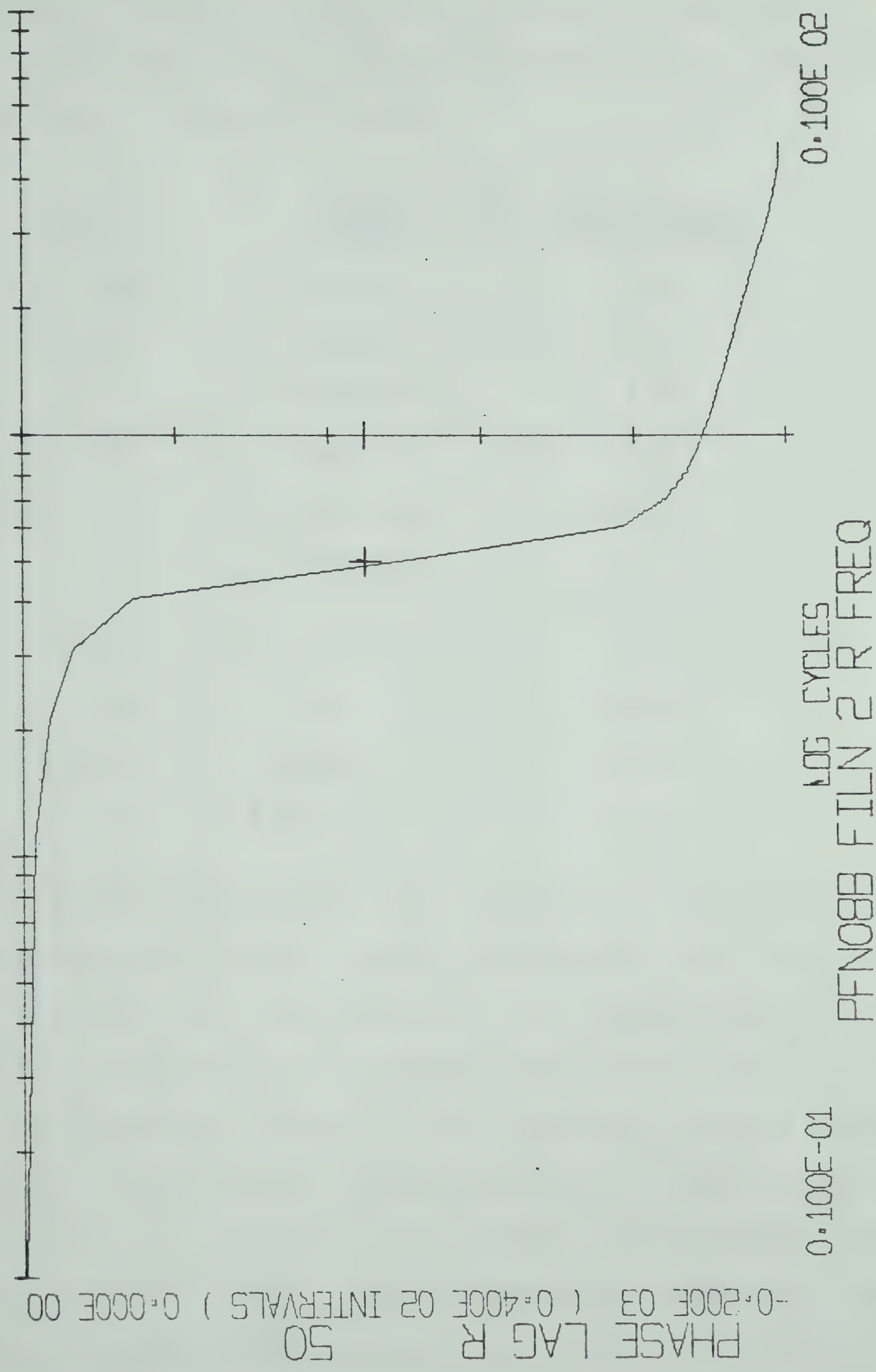

0.000E 00
0.500E 01
(0.500E 00 INTERVALS)
PFTOBC FAST 2 F FREQ
0.000E 00 (0.600E-01 INTERVALS) 0.360E 00
FREQ. CONT. R 150


0.000E 00
(0.250E 02 INTERVALS)
0.250E 03
PTHOBB THRY 2 F TIME
0.400E 01 (0.100E 01 INTERVALS) 0.900E 01
OUT PULSE R 1200


0.100E-01
0.100E 02
LOG CYCLES
PFTOBA FAST 2 F FREQ
0.100E-01 LOG CYCLES 0.100E 02
AMP. RATIO R 150


0.100E-01
0.100E 02
LOG CYCLES
PFTO8B FAST 2 F FREQ
-0.200E 03 (0.400E 02 INTERVALS) 0.000E 00
PHASE LAG R 150


0.100E 02
LOG CYCLES
PFNOBA FILN 2 R FREQ
0.100E-01
AMP. RATIO R
50
0.100E-01
LOG CYCLES
0.100E 02


0.100E-01
0.100E 02
LOG CYCLES
PFNOBB FILN 2 R FREQ
-0.200E 03 (0.400E 02 INTERVALS) 0.000E 00
PHASE LAG R 50

The graphs of the pulse data and the frequency content of the input pulse for each run are presented in Appendix II. The runs which were made can be summarized as follows:

RUN NO.	INPUT PULSE SHAPE	INPUT PULSE DURATION (SEC)
14	Rectangle	1.0
10	Rectangle	3.0
12	Rectangle	6.0
16	Rectangle	6.0
11	Rectangle	10.0
17	Rectangle	15.0
18	Ramp	6.0
13	Ramp	10.0
19	Ramp	15.0
20	Ramp	20.0

The number of data points on the amplitude ratio and phase lag graphs derived using the Fast Fourier Transform (FFT) cover a relatively narrow frequency range. This is related to the sampling interval which affects the frequency range as mentioned in Sections 3.2.4 and 6.2.2. This range is less than adequate to truly characterize the heat exchanger. The problem with the results obtained with the FFT is related to the pulse data sampling interval which was 0.25 seconds. In all likelihood, this interval is far less than that required to adequately characterize the response of the heat exchanger process. If a larger sample interval had been used, the frequency band would have been extended on both ends and the results would have been more useful. However, these graphs do

serve to demonstrate the problem which can be encountered when a pulse test is not properly designed giving consideration to the restrictions imposed by the FFT. It should be emphasized here that improper design implies the FFT yields insufficient, not erroneous, data because the results obtained from the FFT correlate with those obtained using the other two techniques. Due to the limited usefulness of the FFT results, a minimal number of graphs obtained using this technique are included in Section 6.3.2.1.

6.3.1 The Effect of Input Pulse Duration and Shape on Frequency Content

The frequency content curves for the experimental input pulses (flow) are presented in Appendix II. In each case the general shape of the curve was in agreement with the results of pulse shape studies performed by Dreifke (25), Clements (12) and Hougen (40).

For the rectangular shaped pulses the frequency content curves exhibited distinct zeros which occurred at the frequencies predicted by theory (see Section 3.2.2). As examples, consider Graphs PFN16C, PFN11C and PFN17C in Appendix II, where the input pulse durations were 6, 10 and 15 seconds, respectively. Theoretically, the first zero in the frequency content curve should occur at the frequency when $w \cdot d = 2\pi$, which for these examples would be 1.05, 0.628 and 0.419 radians per second, respectively. A study of the graphs reveals that the frequencies at the zeros agree with these values within a reasonable experimental error. Also, it should be noted that the second and third zeros, where plotted, occur at frequencies which are multiples of the frequency value at the first zero. This also agrees with the theory in Section 3.2.2.

For the ramped shaped pulses the frequency content curves were more difficult to compare with the theory on a quantitative basis because they do not exhibit zeros. Therefore, the validity of the curves obtained in this work was based upon a general agreement with the results reported by other workers (12, 25, 40). That is, the curves monotonically decreased in amplitude with increasing frequency but never actually reach zero amplitude, and the rate of decay in amplitude became greater as the pulse duration became longer. These characteristics are exhibited in the frequency content curves in Appendix II for runs 18, 19 and 20.

6.3.2 A Discussion of the Amplitude Ratio and Phase Lag of the Test Heat Exchanger

In general, the amplitude ratio and phase lag graphs obtained from the test heat exchanger were noisy, so that continuous line graphs were somewhat obscure. It was found that plotting the data in point form rendered the graphs more meaningful and this procedure was used for all the experimental amplitude ratio and phase lag graphs.

Owing to time limitations, under which the experimental phase of this work had to be conducted, no repeated runs were possible. A total of ten pulse tests (runs) were conducted over a two-day period and the data obtained were analyzed in the following week. Because of the conditions for testing, the discussion and conclusions presented in the following text should be further substantiated before they are used as a strong guideline in future work.

Runs 12 and 16 were the only two runs for which the pulse duration and shape were the same, (six second-rectangular). Run 13 was, in the author's opinion, for reasons given later, the pulse test which yielded the most valid frequency response data.

In order to evaluate the frequency response results obtained, it was necessary to establish the low frequency or steady state gain of the test heat exchanger for the operating point at which the pulse tests were conducted. If it is assumed that the first half of the long duration rectangular pulse used in run 17 can be considered as a step, then (from data in Table 6) the steady state gain of the heat eschanger was approximately 2 04/3.38 = 0 603. This means that Graphs PTP14A, PTP01A and PFN18A are all in error because in each case an amplitude ratio of greater than one is demonstrated. It is interesting to note that in each of these tests the pulse duration was short relative to other pulses of the same shape. It is the author's opinion that the erroneous nature of these runs was to some extent related to an insufficient driving force on the input pulse as discussed in Section 3.2.2.

Also, in reference to short duration input pulses, it is interesting to note the "noisy" nature of the frequency content curves in Graphs PFN14C and PFT14C (Appendix II) which were derived from the one second duration rectangular input pulse. It is possible that this "noise" was a result of the lower signal to noise ratio which exists for shorter duration time pulses

As mentioned previously, it is the author's opinion that the results from all runs are invalid with the possible exception of run 13. This opinion is based upon the fact that all experimental runs exhibited a significant error of closure on both the input and output pulse It has been demonstrated in this work and reported by others (Section 6 2.4) that an error of closure on the output pulse alone tends

to render the amplitude ratio and phase lag graphs unreliable In the example shown in Graph PFN03W on page 172 (discussed in Section 6.2.2), an error of closure of 10 7% on the output pulse rendered the amplitude ratio curve unreliable. From a study of the errors of closure on the output pulse alone, only Graph PTP17A, PFN18A, PFN13A and PFT16A can be considered to contain valid information. Examination of these graphs lead to the following comments:

a. Graph PTP17A - The data points are badly scattered and the location of the amplitude ratio curve can not be readily determined. Some of the scatter could be due to error of closure and the rest could be attributed to the zeros in the frequency content curve of the input pulse. These occur at frequencies of approximately 0.875, 1.75, 2.625 and 3.5 radians per second

b. Graph PFN18A - The run can be declared invalid on the basis of a study of pulse data in Graphs PEX18A and PEX18B in Appendix II. It can be seen that the system was not at steady state when the pulse was introduced, in fact the temperature was dropping From the information under item (f) in Section 6.3 2.1, it is apparent that the maximum temperature change which occurred was excessively large in comparison with data obtained from runs 13, 19 and 20 in which the input pulse durations were longer than that used in run 18. (All these runs involved ramp pulses.) On the basis of this information it is reasonable to suggest that the pulse in run 18 served to amplify the already decreasing temperature so that a greater than normal temperature deviation occurred. It

is possible that the amplitude ratio of greater than one in Graph PFN18A is due to this condition as well as to the short pulse duration as discussed previously.

c. Graph PFN13A - The data points are not too severely scattered and the error of closure on the output pulse is the smallest of all the experimental runs. This appears to be valid.

d. Graph PFT16A - The results which cover a narrow frequency range were obtained from the Fast Fourier transform (Section 6.3). The narrow range makes it virtually impossible to establish the validity of this graph.

It should be pointed out that the value of the error of closure, as defined in this work (Section 6.3.2.1), is totally dependent upon the value of the last pulse data point selected in the calculation of the Fourier transform. This means that a large error of closure could result when, in fact, the noisy time data has a mean value that represents a well-closed pulse. For example, the error of closure for Graph PFN13A is less than that for Graph PFT13A, although the same pulse data were used in both cases. Therefore, in the above analysis, it was necessary to apply a weighting factor to the relative quality of the steady state data obtained after the pulse (Appendix II). It appears that the steady state data in from run 13 is superior to the data from all other runs which serves to further substantiate the validity of run 13.

On the basis of the above discussion, it seems reasonable to conclude that the error of closure is a dominant factor in results obtained from all the experimental runs. It is the author's opinion that the majority of data scatter on the frequency response results would disappear if the error of closure was minimized or eliminated.

Because of the poor quality of the data obtained in the experimental runs, it was not possible to draw any conclusions regarding the relative merits of the two pulse shapes used. Theoretically the ramp pulse should be better because it does not exhibit zeros in its frequency content curve.

6.3.2.1 Amplitude Ratio and Phase Lag Graphs Derived by Pulse Testing the Test Heat Exchanger

General comments:

a. Table 8 describes the graphs presented on pages 209 - 238 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs was given in Section 6.1.

c. The data presented in this section was derived from the time domain pulse data presented in Appendix II. Three Fourier transform techniques were used in the derivation:

   1. Fast Fourier transform (FT)

   2. Trapezoidal quadrature (TP)

   3. Filon's quadrature (TP)

d. For the results obtained using FT, the sample interval on the time data was 0 025 seconds. For the results obtained using TP or FN, the interval was 0.1 seconds because of the use of subroutine REDPT, Section 5.4.2.3. The effect of REDPT on the data can be seen by comparing Graph PEX13A with Graph PEX13C (Appendix II) where the data for Graph PEX13C was derived by selecting every fourth point from the data plotted in Graph PEX13A. A similar comparison can be made between Graphs PEX13B and PEX13D

e. Error of closure is defined as the difference between the average steady state before the occurrence of the pulse and the last data value used in the calculation of the Fourier transform. The percentage error of closure is calculated by dividing the error of closure by the maximum deviation of the variable from the average steady state during the pulse, i.e., the maximum change in the variable during the pulse. In these cases, the averaged steady state value was the steady state calculated by programs FREQ or KLTLY (Section 5.4.2.3). The value of the pulse curve at the point of maximum deviation was read from the pulse data plots presented in Appendix II. The actual values used to determine the errors of closure are given in Tables 6 and 7.

f. The values used for the steady states and maximum deviations are listed in Table 6. Each set of values in this table is related to an amplitude ratio and phase lag graph presented on pages 209 - 238. A set of values applies to a graph when the Fourier transform technique and the run number are the same as the Identification designation in Table 6.

g. The values used for the errors of closure are listed in Table 7. The relationship between the Identification designation in Table 7 and the graphs presented in this section is as outlined under item (f).

h. All amplitude ratio graphs are on the same logarithmic-logarithmic scale to facilitate comparisons. A single exception is Graph PFN13D which has a different abscissa scale.

TABLE 6

INFORMATION ON THE INPUT AND OUTPUT
PULSES ASSOCIATED WITH EXPERIMENTAL RUNS

	INPUT PULSE (GPM (US))			OUTPUT PULSE (^{o}F)		
IDENTIFICATION	STEADY STATE	PEAK (MAX.)	MAXIMUM DEVIATION	STEADY STATE	DIP (MAX.)	MAXIMUM DEVIATION
TP14	8.13	11.40	3.27	136.99	136.10	0.89
TP10	8.13	12.10	3.97	137.94	136.36	1.58
TP12	8.16	11.88	3.72	138.01	135.50	2.51
TP16	8.12	11.51	3.39	137.42	135.28	2.14
TP11	8.18	11.91	3.73	137.67	135.16	2.51
TP17	8.16	11.54	3.38	137.64	135.60	2.04
FN18	8.25	11.90	3.65	137.91	135.37	2.54
FN13	8.10	12.05	3.95	137.86	136.20	1.66
FN19	8.10	11.50	3.40	137.33	135.36	1.97
FN20	8.12	11.49	3.37	137.90	135.09	2.81
FT14	8.12	11.40	3.28	136.98	136.10	0.88
FT12	8.09	11.88	3.79	138.07	135.50	2.57
FT16	8.12	11.51	3.39	137.42	135.28	2.14
FT17	8.16	11.54	3.38	137.64	135.60	2.04
FT13	8.08	12.05	3.97	137.86	136.20	1.66

TABLE 7

ERRORS OF CLOSURE ON INPUT AND OUTPUT PULSES FOR EXPERIMENTAL RUNS

IDENTIFICATION	INPUT PULSE (GPM US) ERROR OF CLOSURE	INPUT PULSE (GPM US) PERCENT ERROR OF CLOSURE	OUTPUT PULSE (^oF) ERROR OF CLOSURE	OUTPUT PULSE (^oF) PERCENT ERROR OF CLOSURE
TP14	0.0871	2.67	0.197	22.2
TP10	0.0742	1.87	0.249	15.8
TP12	0.0910	2.45	0.156	6.22
TP16	0.00253	0.0746	0.249	11.60
TP11	0.174	4.80	0.309	12.3
TP17	0.0732	2.16	0.0619	3.03
FN18	0.0474	1.30	0.0244	0.96
FN13	0.0544	1.75	0.0136	0.82
FN19	0.156	4.60	0.380	19.30
FN20	0.0347	1.03	0.317	11.25
FT14	0.0229	0.70	0.106	12.05
FT12	0.0256	0.68	0.227	8.84
FT16	0.0748	2.20	0.057	2.66
FT17	0.0732	2.16	0.334	16.2
FT13	0.0734	1.85	0.141	8.5

i. All phase lag graphs are on the same semi-logarithmic scale to facilitate comparisons.

j. Under the "Input Pulse Shape" column of Table 8, "RE" implies a rectangular pulse and "RA" a ramp pulse.

k. The graphs on pages 209 - 238 are presented in the following general order which is not necessarily in accordance with run number:

1. amplitude ratio calculated by:

   (i) TP using:

   - data from pulse tests in order from the narrowest to the widest rectangular shaped input pulse

   (ii) FN using:

   - data from pulse tests in order from the narrowest to the widest ramp shaped input pulse

   (iii) FT using:

   - as under (i) and (ii) above but with a reduced number of cases considered for each input pulse shape

2. phase lag calculated by:

   (i), (ii) and (iii) as indicated in (1).

6.3.3 The Effect of Time and Spectral Smoothing on the Experimental Results

Using the data from run 13, a limited investigation was conducted to determine whether time and/or spectral smoothing would reduce the "noise" in the amplitude ratio graphs.

TABLE 8

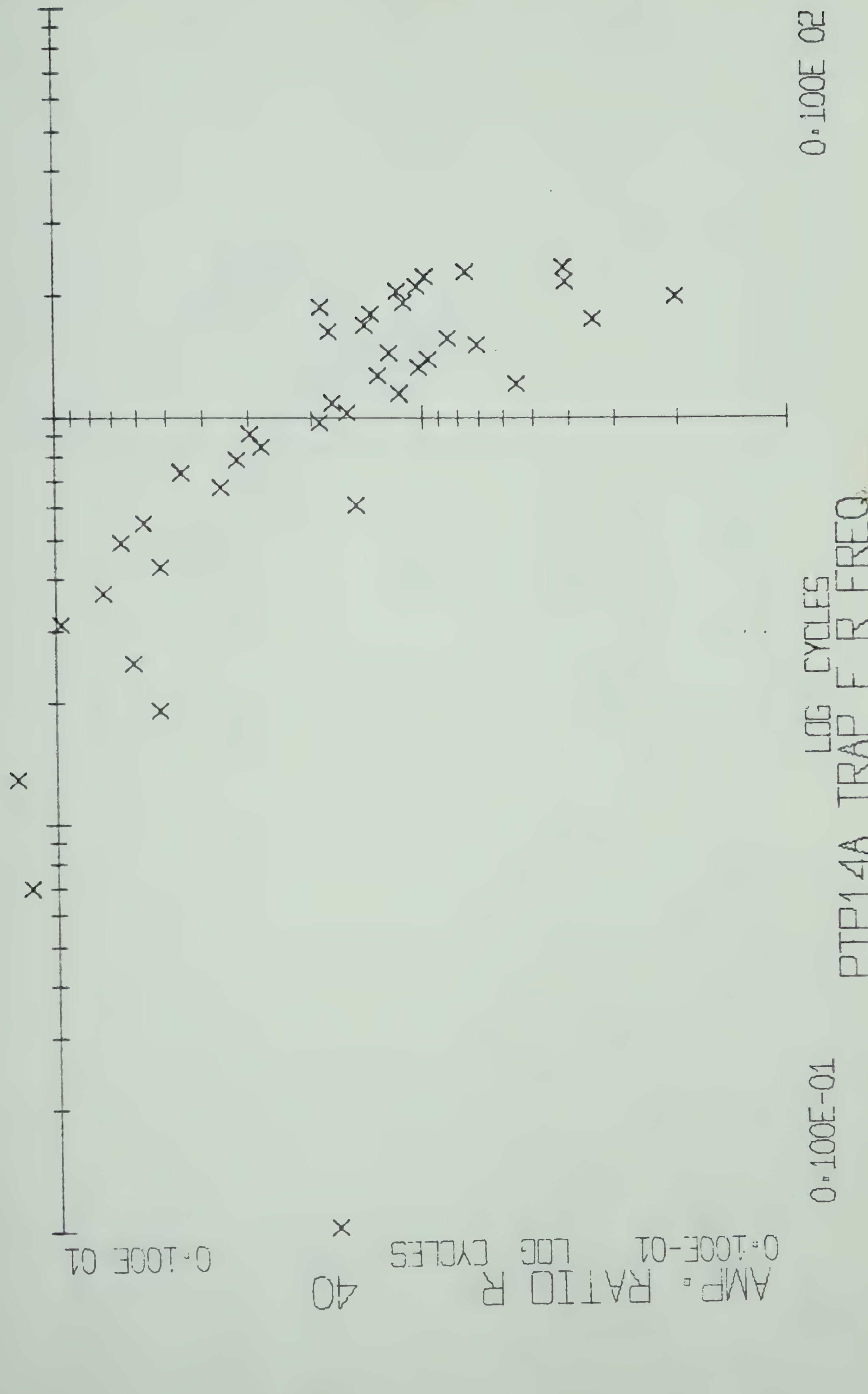
LIST OF FREQUENCY RESPONSE GRAPHS FOR THE TEST HEAT EXCHANGER

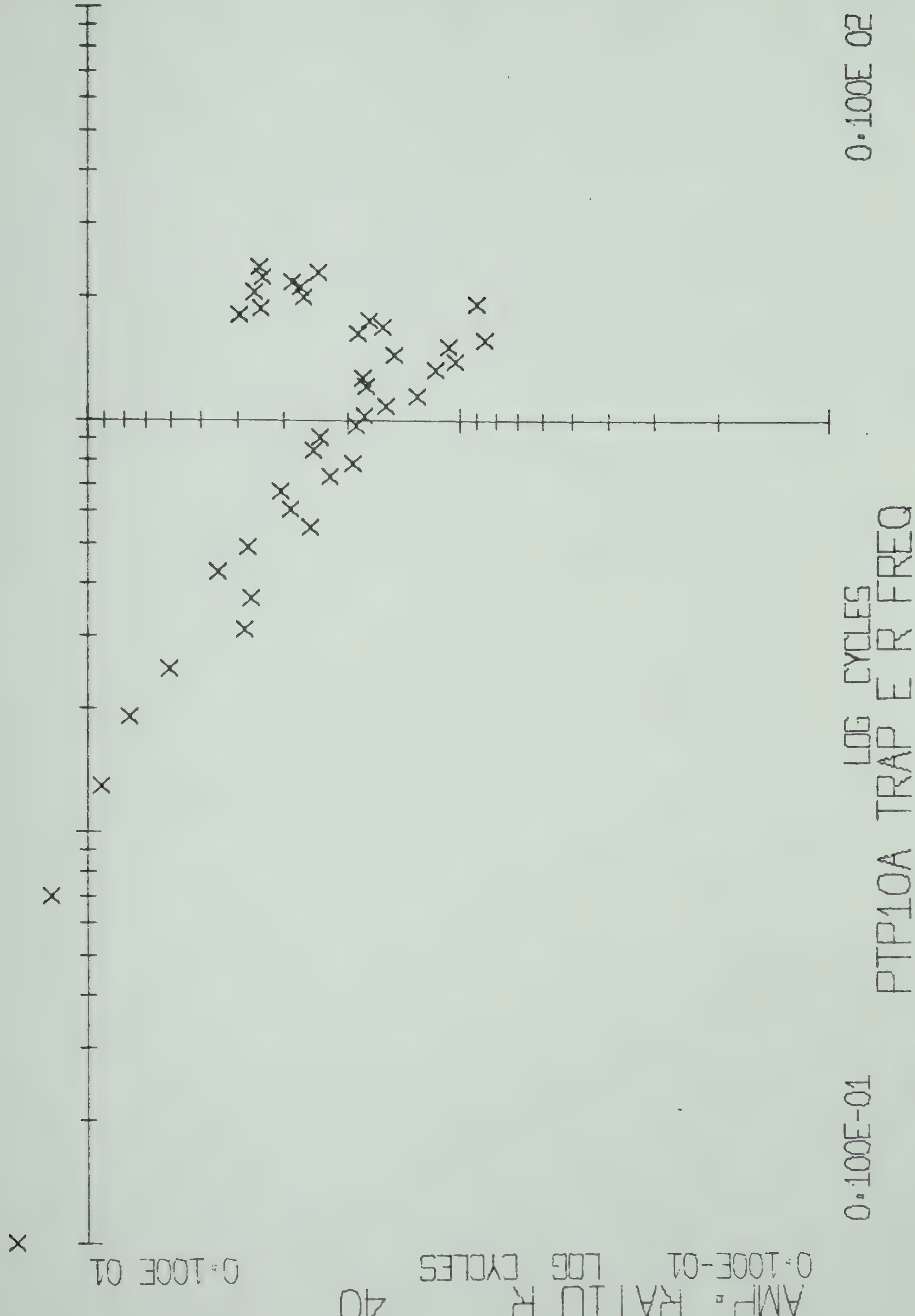
GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	INPUT PULSE SHAPE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PTP14A	209	1.0	RE	TP	Amplitude ratio
PTP10A	210	3.0	RE	TP	Amplitude ratio
PTP12A	211	6.0	RE	TP	Amplitude ratio
PTP16A	212	6.0	RE	TP	Amplitude ratio
PTP11A	213	10.0	RE	TP	Amplitude ratio
PTP17A	214	15.0	RE	TP	Amplitude ratio
PFN18A	215	6.0	RA	FN	Amplitude ratio
PFN13D	216	10.0	RA	FN	Amplitude ratio
PFN19A	217	15.0	RA	FN	Amplitude ratio
PFN20A	218	20.0	RA	FN	Amplitude ratio
PFT14A	219	1.0	RE	FT	Amplitude ratio
PFT12A	220	6.0	RE	FT	Amplitude ratio
PFT16A	221	6.0	RE	FT	Amplitude ratio
PFT17A	222	15.0	RE	FT	Amplitude ratio
PFT13A	223	10.0	RA	FT	Amplitude ratio
PTP14B	224	1.0	RE	TP	Phase lag
PTP10B	225	3.0	RE	TP	Phase lag
PTP12B	226	6.0	RE	TP	Phase lag
PTP16B	227	6.0	RE	TP	Phase lag
PTP11B	228	10.0	RE	TP	Phase lag

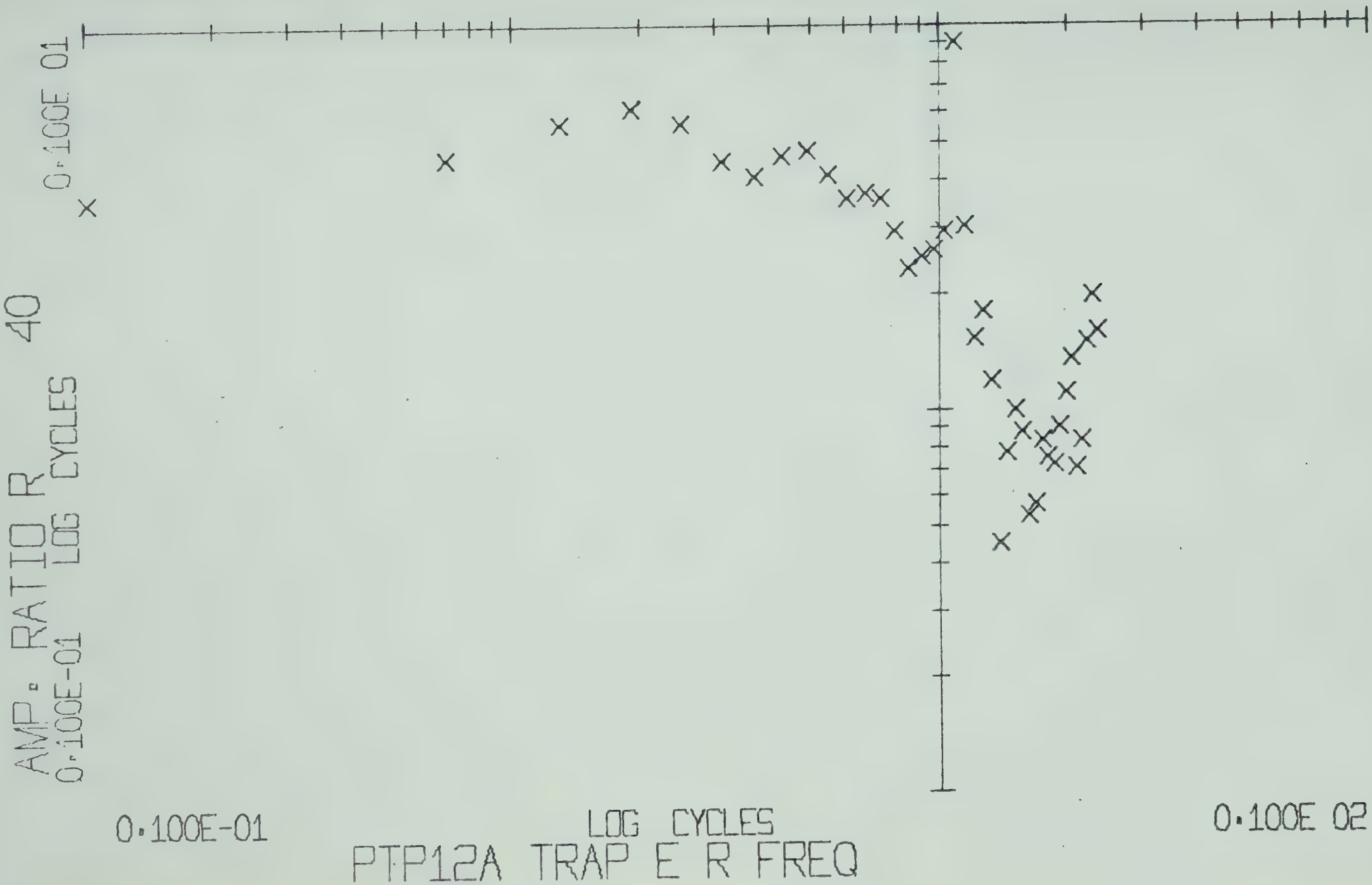
TABLE 8 (CONT'D)

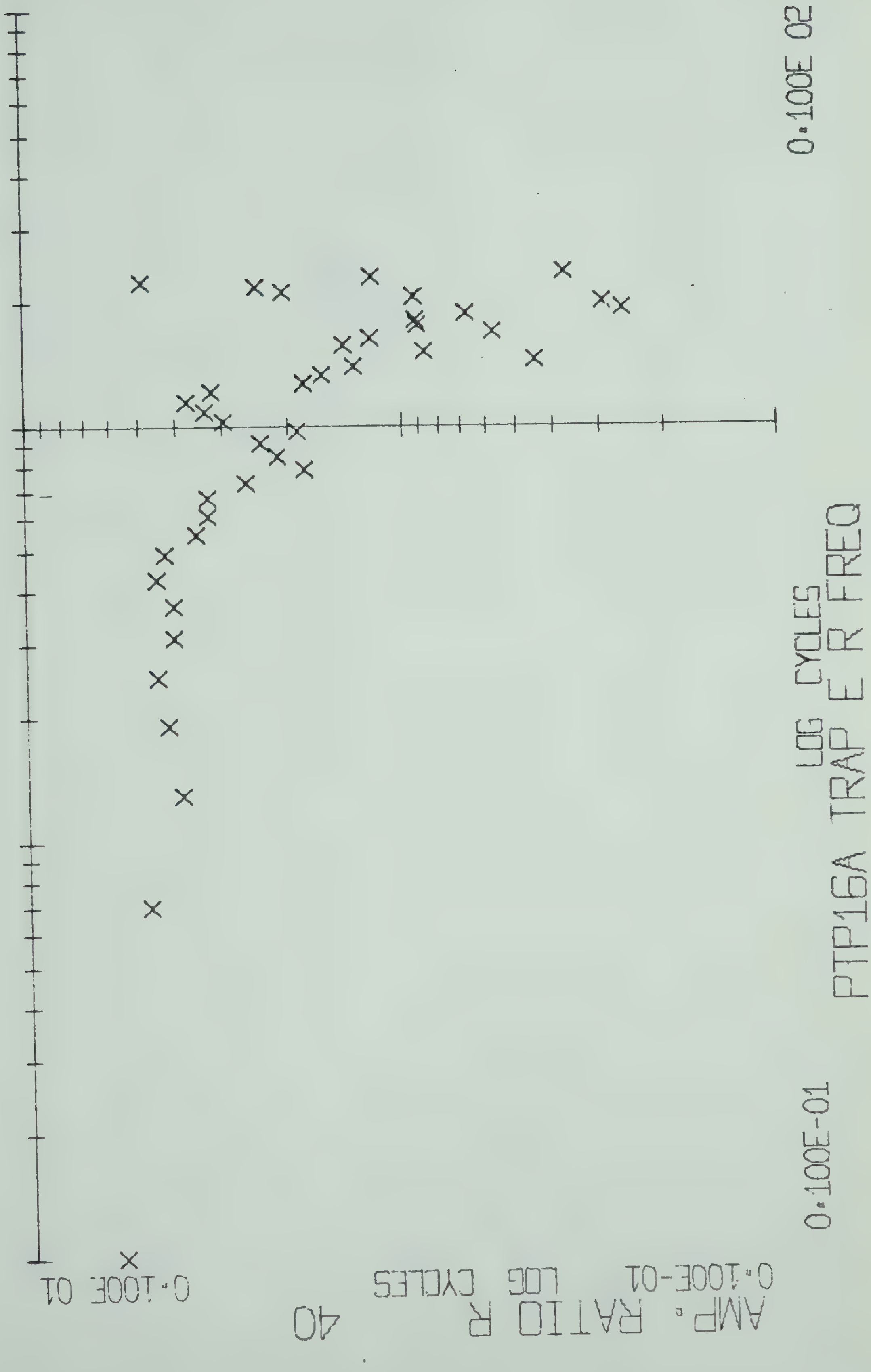
LIST OF FREQUENCY RESPONSE GRAPHS
FOR THE TEST HEAT EXCHANGER

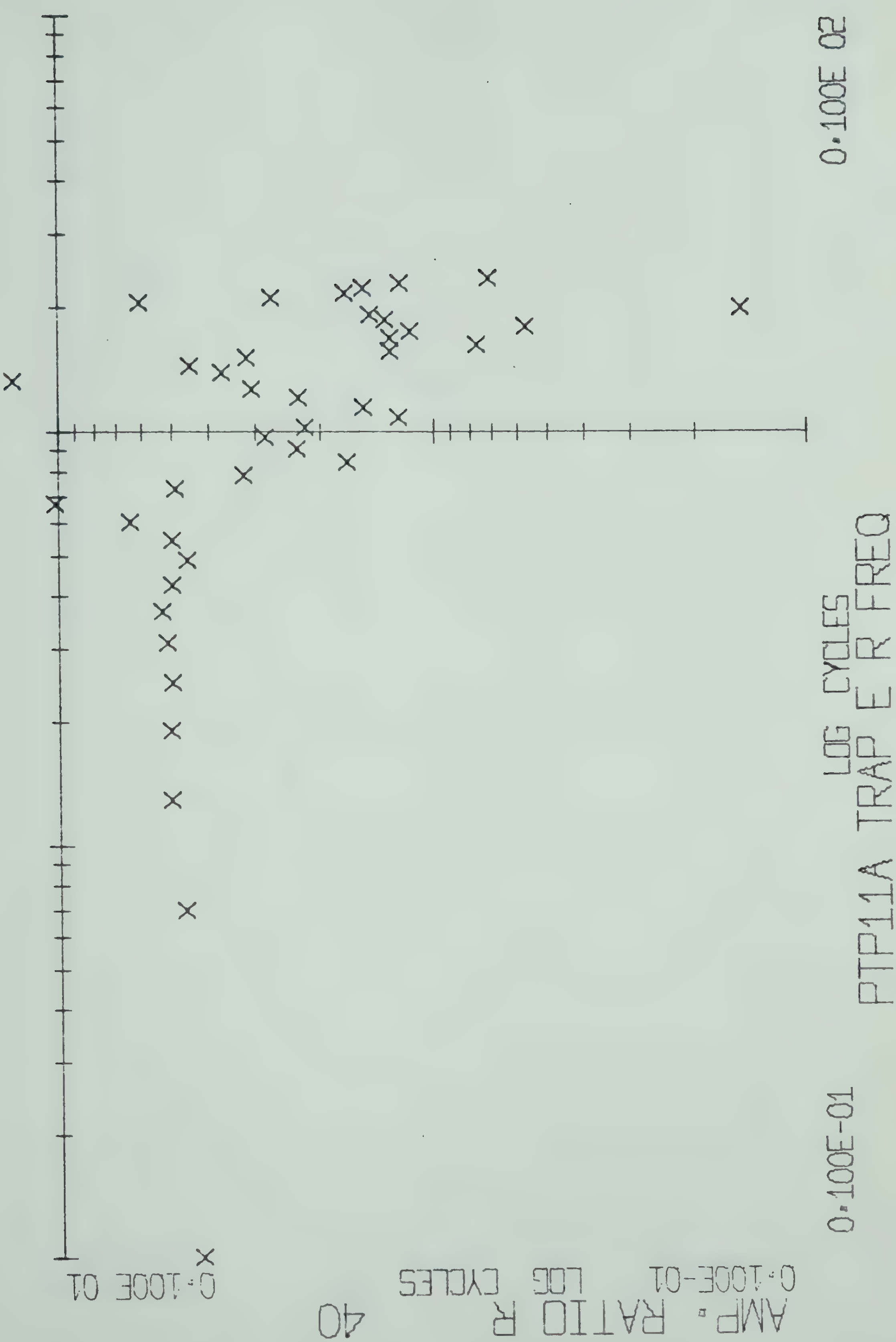
GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	INPUT PULSE SHAPE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PTP17B	229	15.0	RE	TP	Phase lag
PFN18B	230	6.0	RA	FN	Phase lag
PFN13B	231	10.0	RA	FN	Phase lag
PFN19B	232	15.0	RA	FN	Phase lag
PFN20B	233	20.0	RA	FN	Phase lag
PFT14B	234	1.0	RE	FT	Phase lag
PFT12B	235	6.0	RE	FT	Phase lag
PFT16B	236	6.0	RE	FT	Phase lag
PFT17B	237	15.0	RE	FT	Phase lag
PFT13B	238	10.0	RA	FT	Phase lag


0.100E 02
0.100E-01
LOG CYCLES
PTP14A TRAP E R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

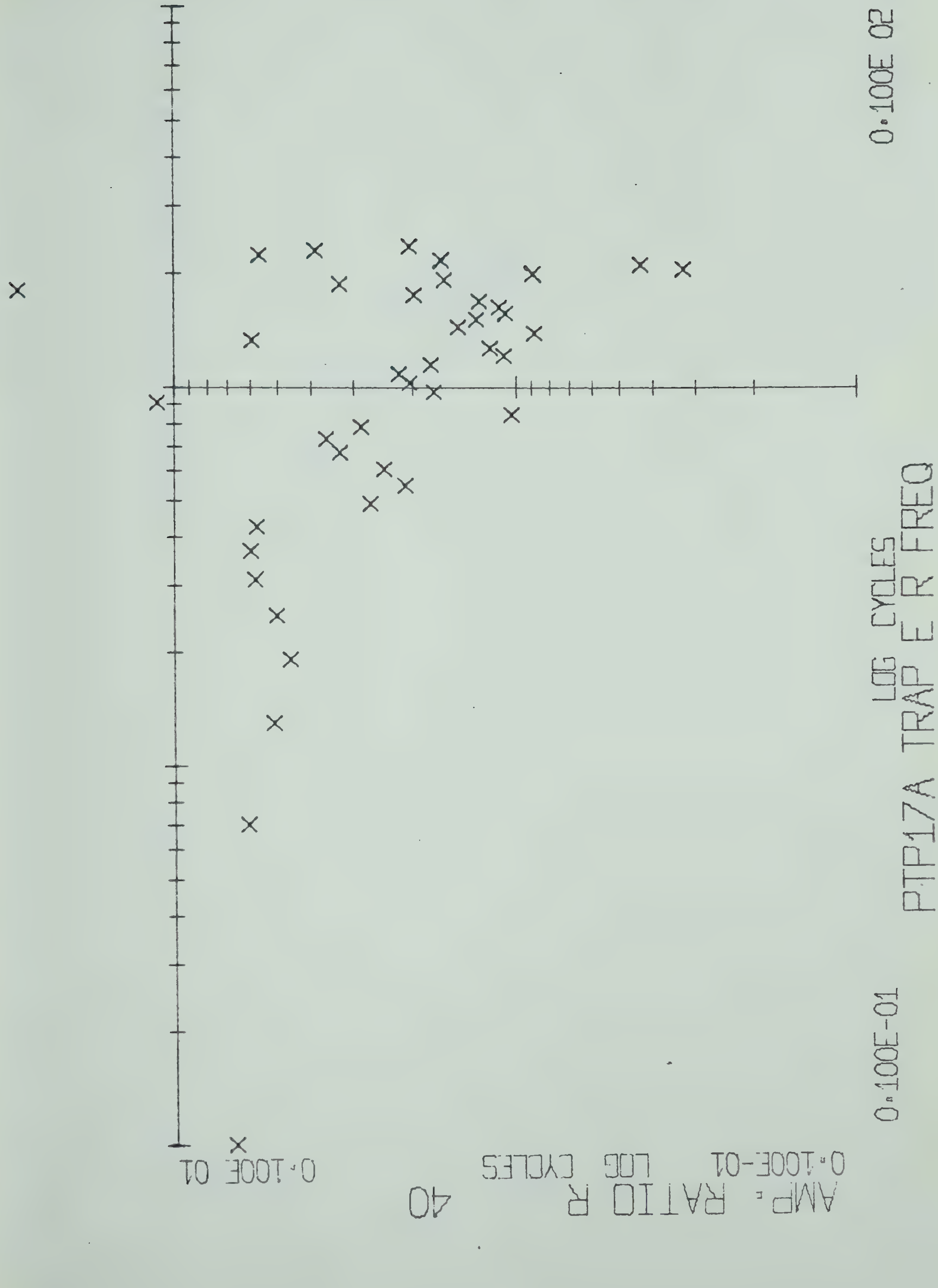

0.100E 02
LOG CYCLES
PTP10A TRAP E R FREQ
0.100E-01
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

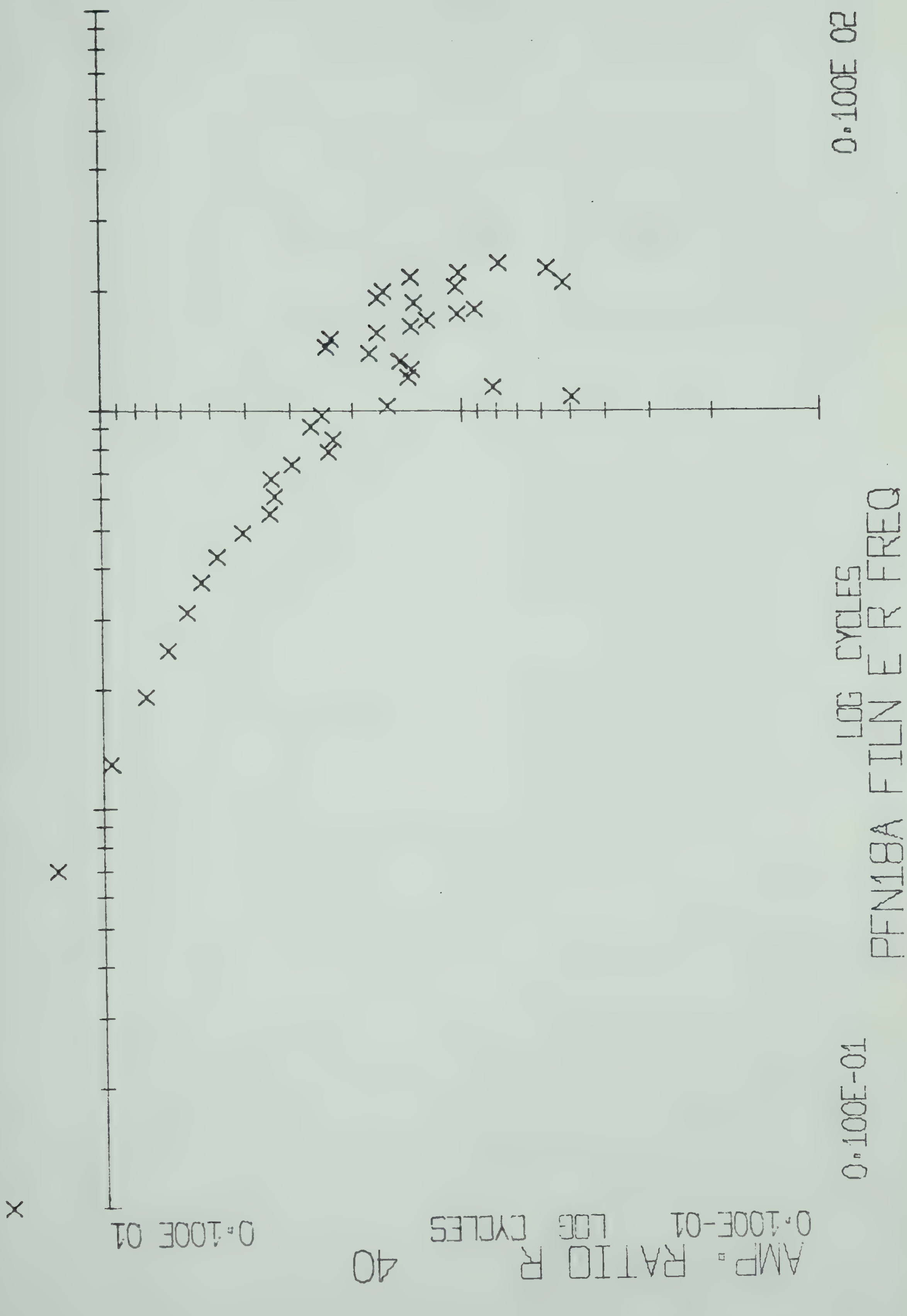

AMP. RATIO R 40
0.100E-01 LOG CYCLES
0.100E 01
0.100E-01
LOG CYCLES
0.100E 02
PTP12A TRAP E R FREQ

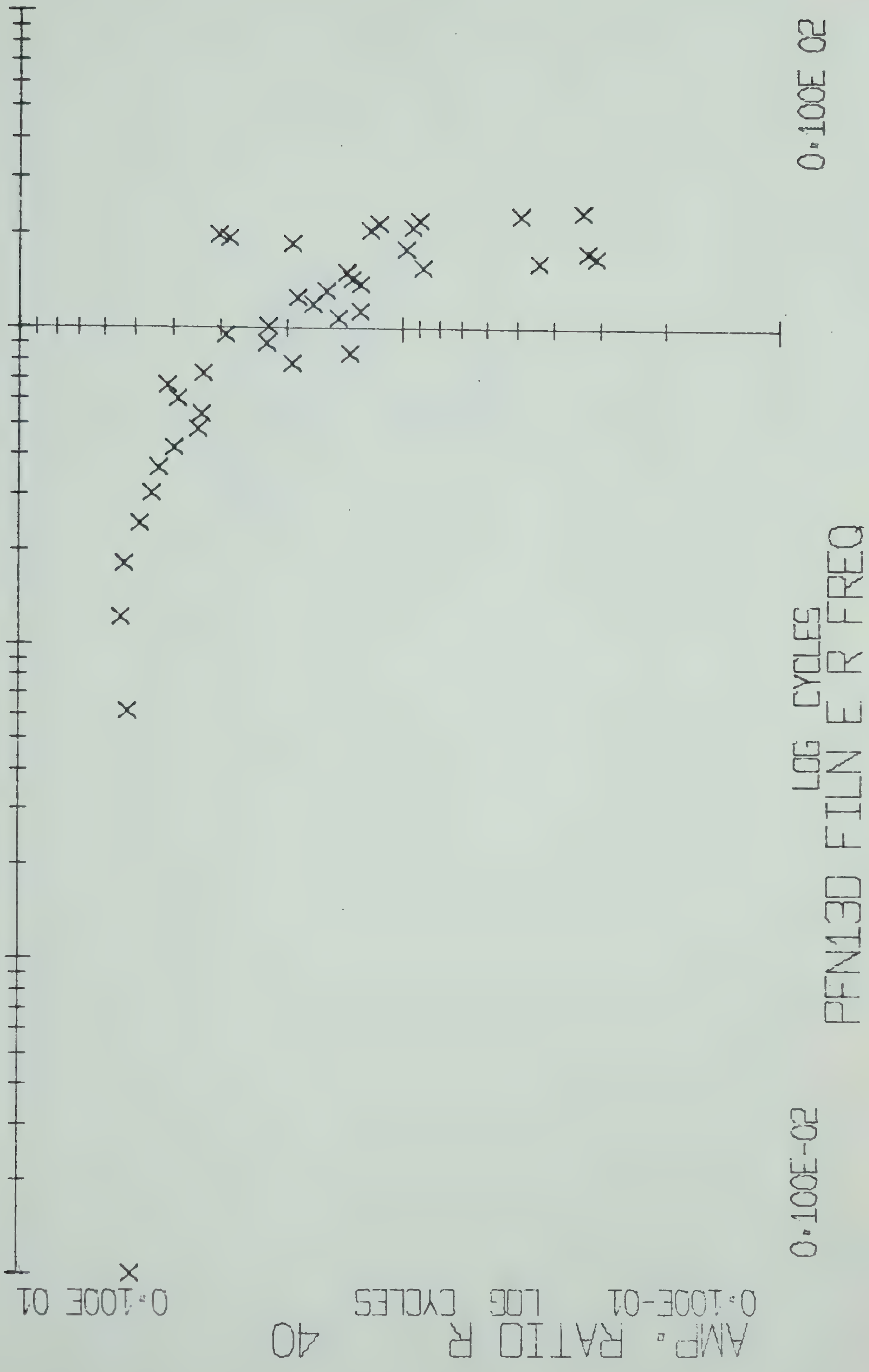

0.100E 02
0.100E-01
LOG CYCLES
PTP16A TRAP E R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

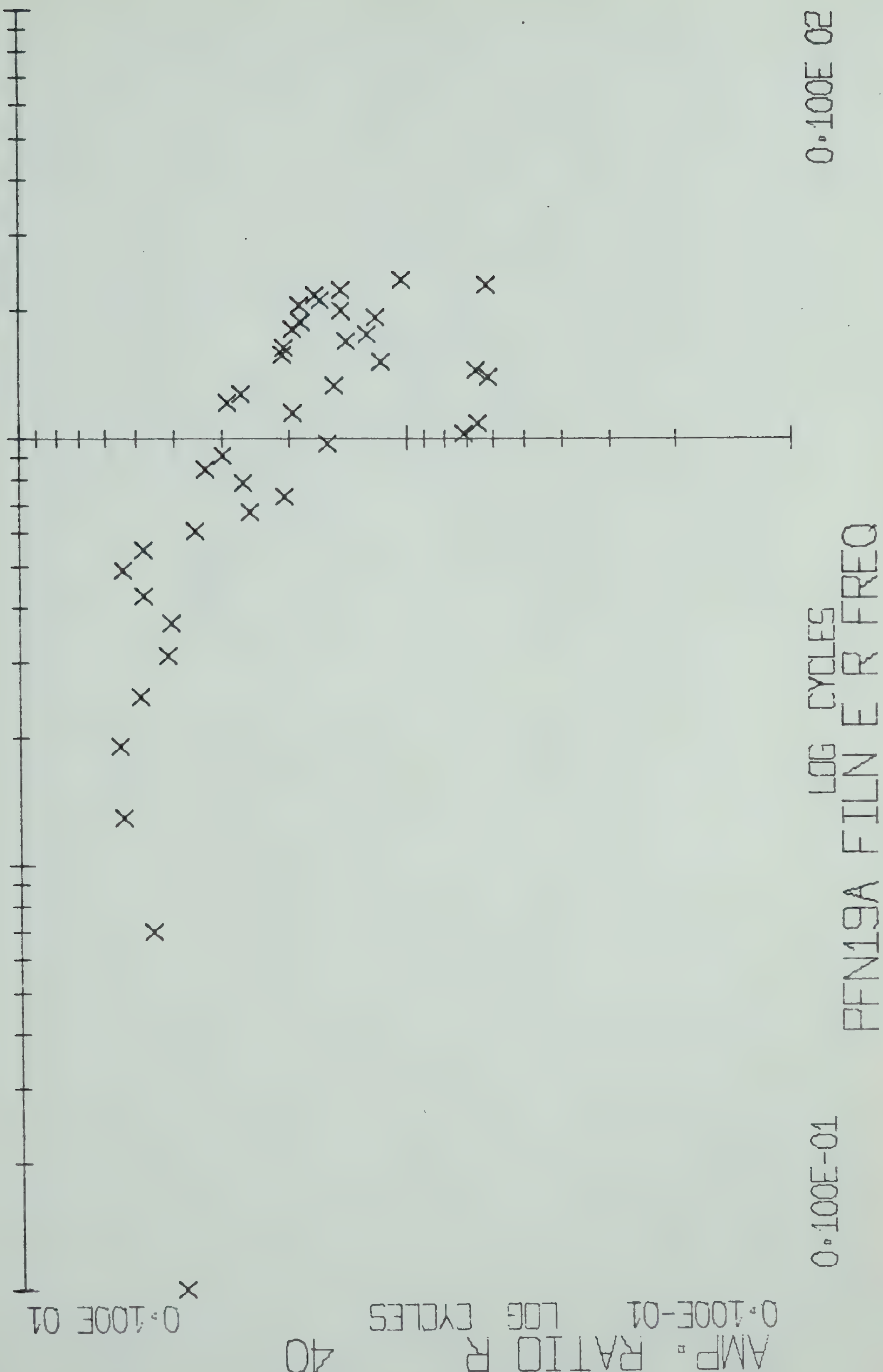


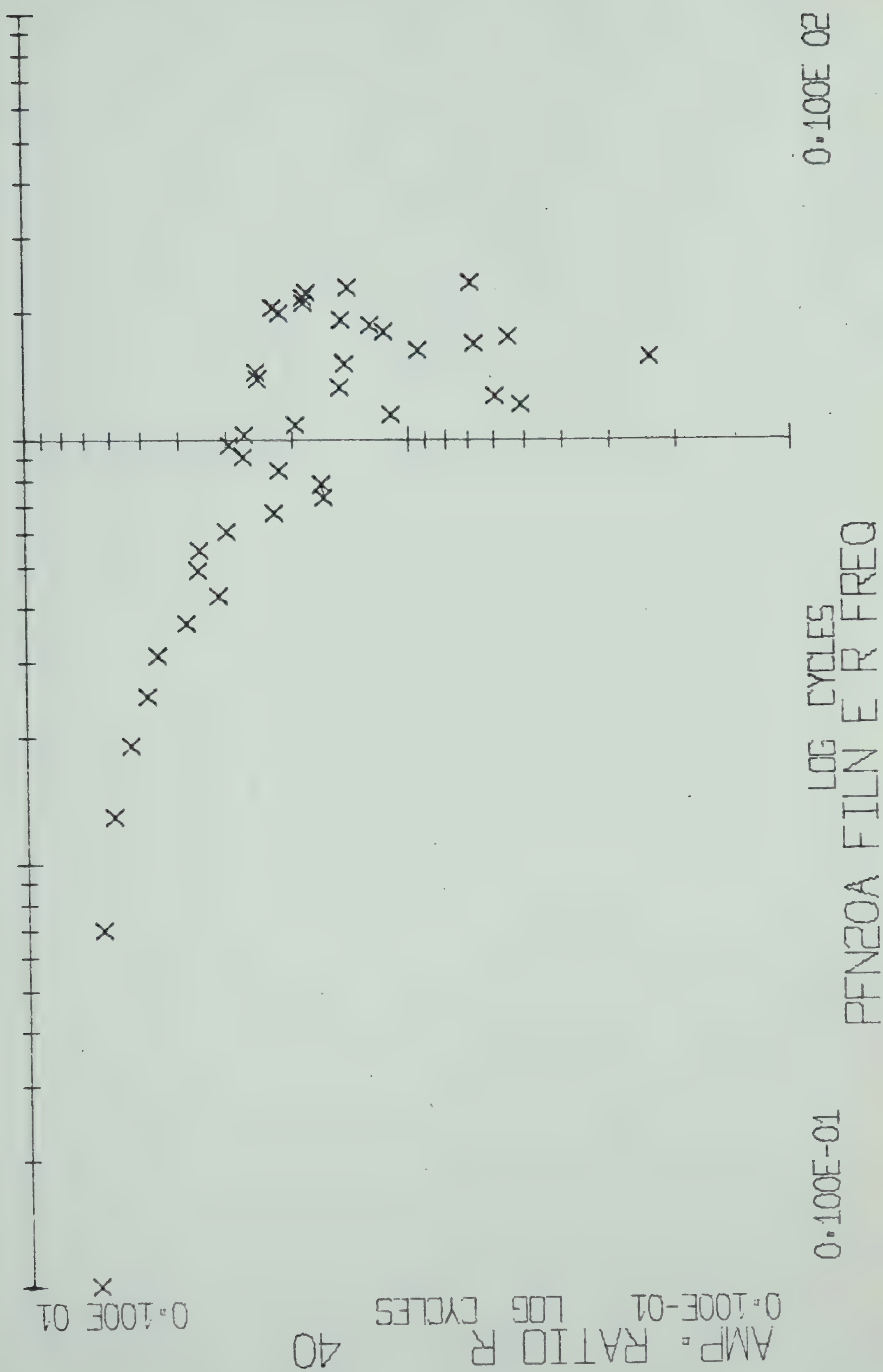
0.100E 02
LOG CYCLES
PTP11A TRAP E R FREQ
0.100E-01
0.100E 01
LOG CYCLES
0.100E-01
AMP. RATIO R
40

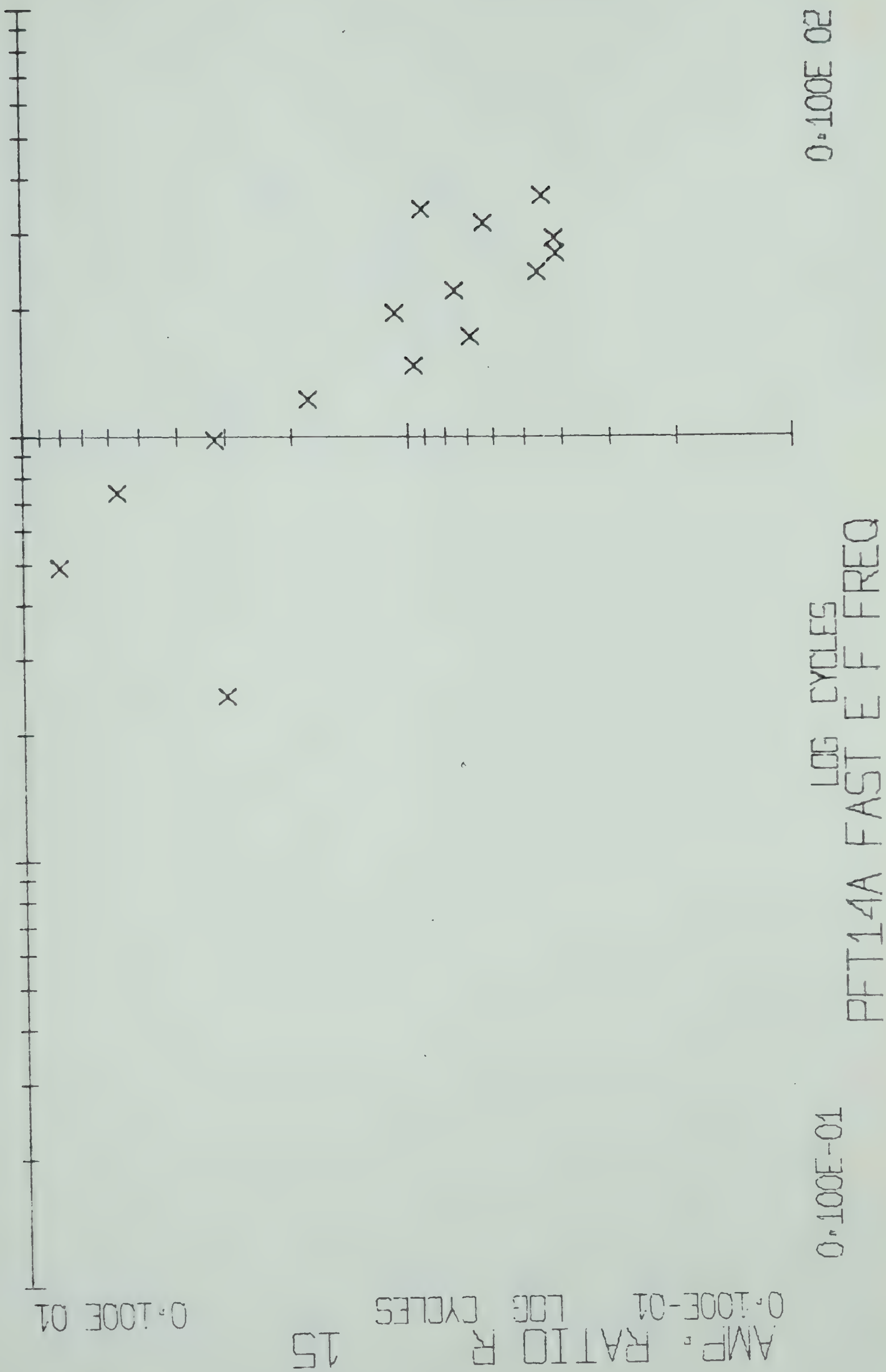

0.100E 02
0.100E-01
LOG CYCLES
PTP17A TRAP E R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

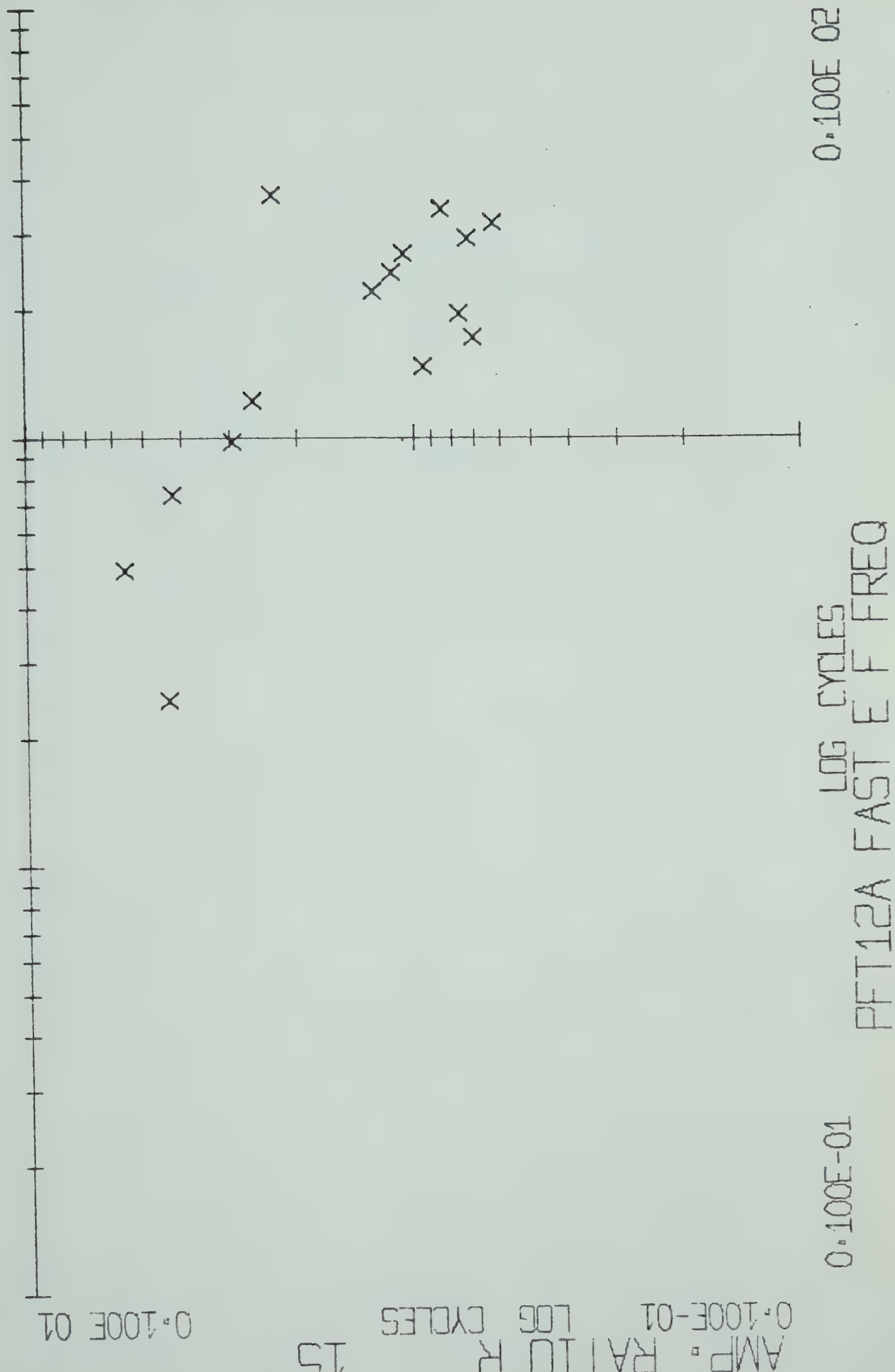

0.100E 02
0.100E-01
LOG CYCLES
PFN18A FILN E R FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

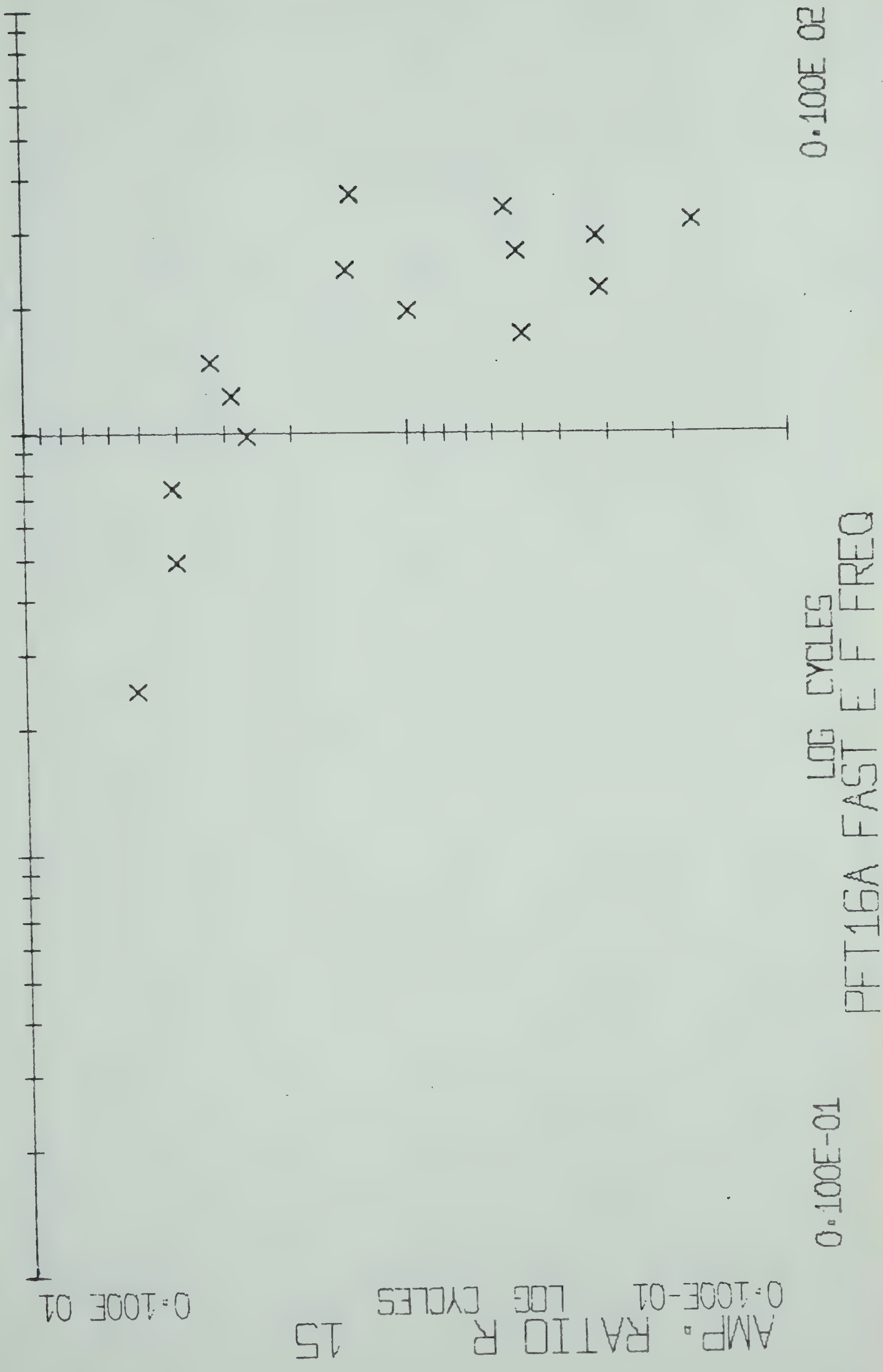

0.100E 02
LOG CYCLES
PFN13D FILN E R FREQ
0.100E-02
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 40

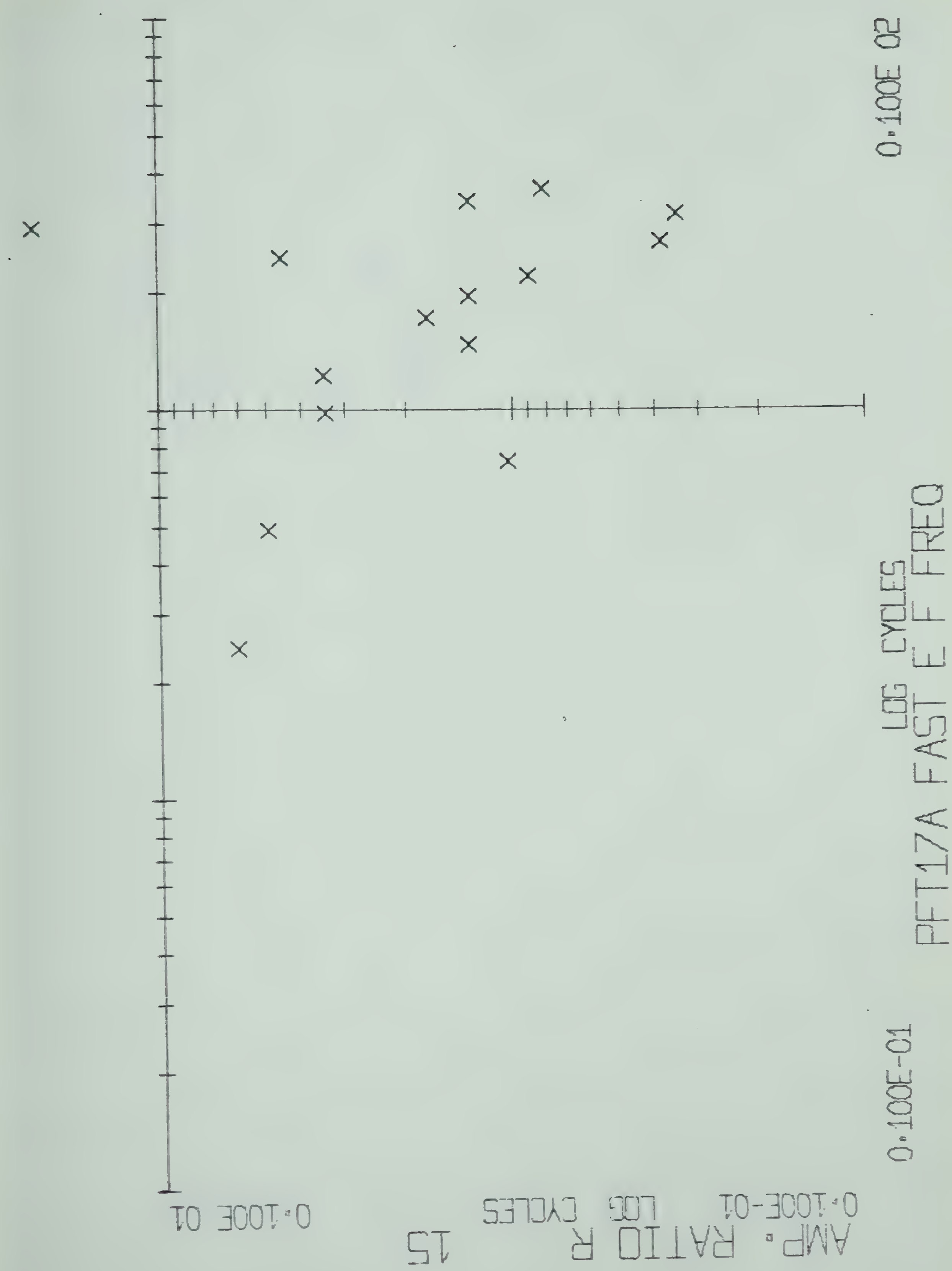

0.100E 02
0.100E-01
LOG CYCLES
PFN19A FILN E R FREQ
AMP. RATIO R 40
0.100E-01 LOG CYCLES 0.100E 01

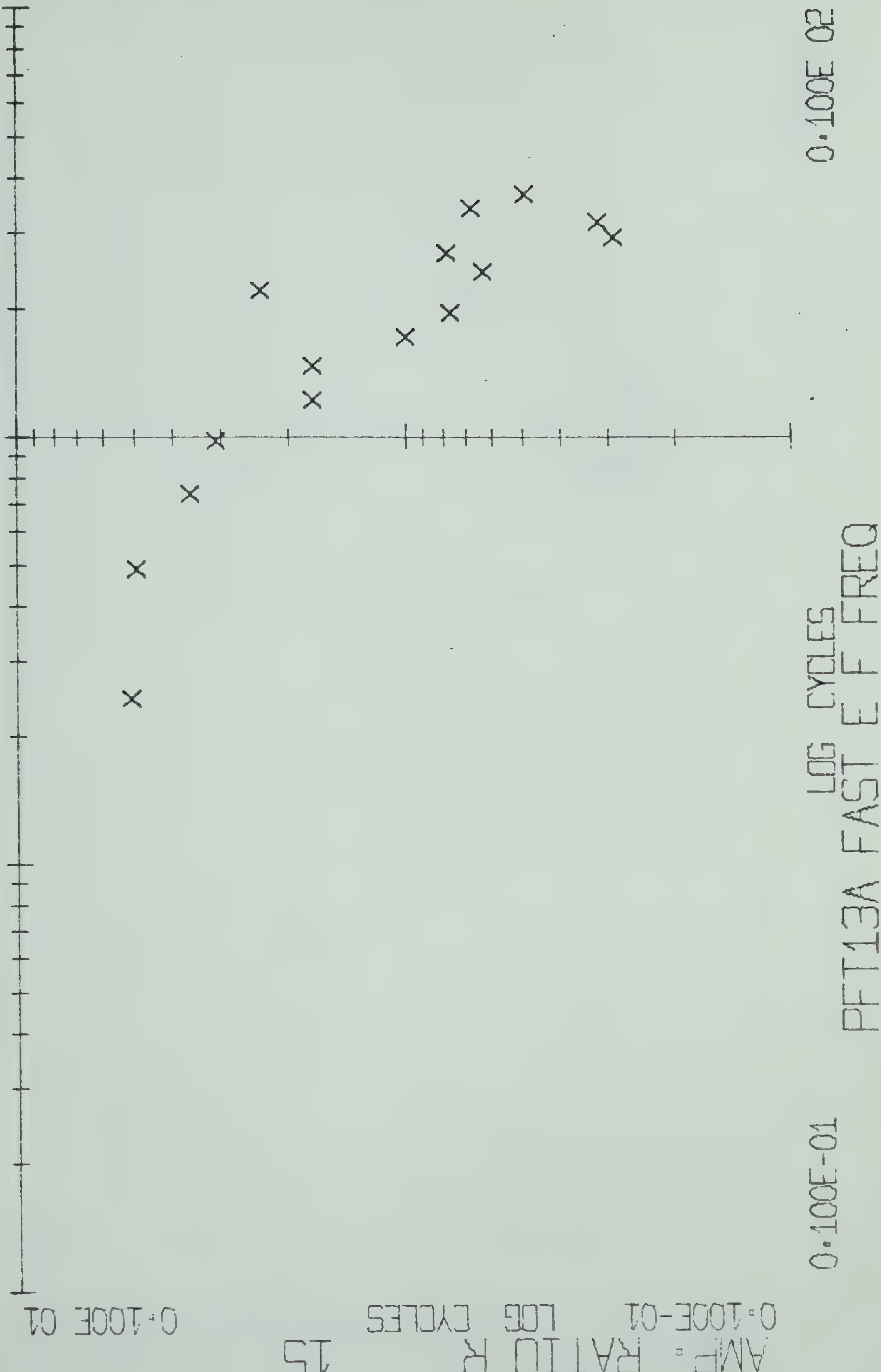

0.100E 02
LOG CYCLES
PFN20A FILN E R FREQ
0.100E-01
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
40

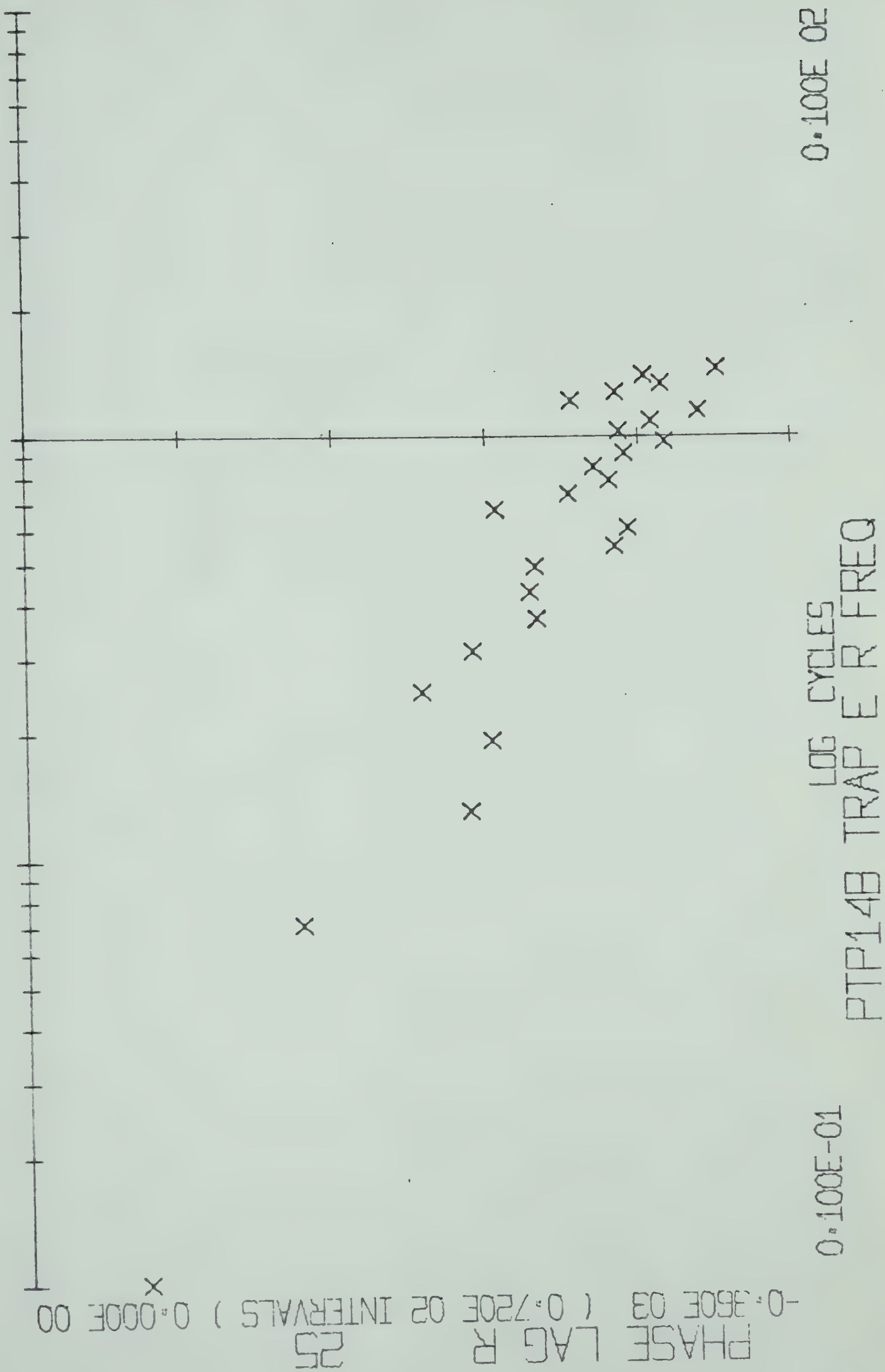

0.100E 02
0.100E-01
LOG CYCLES
PFT14A FAST E F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R 15

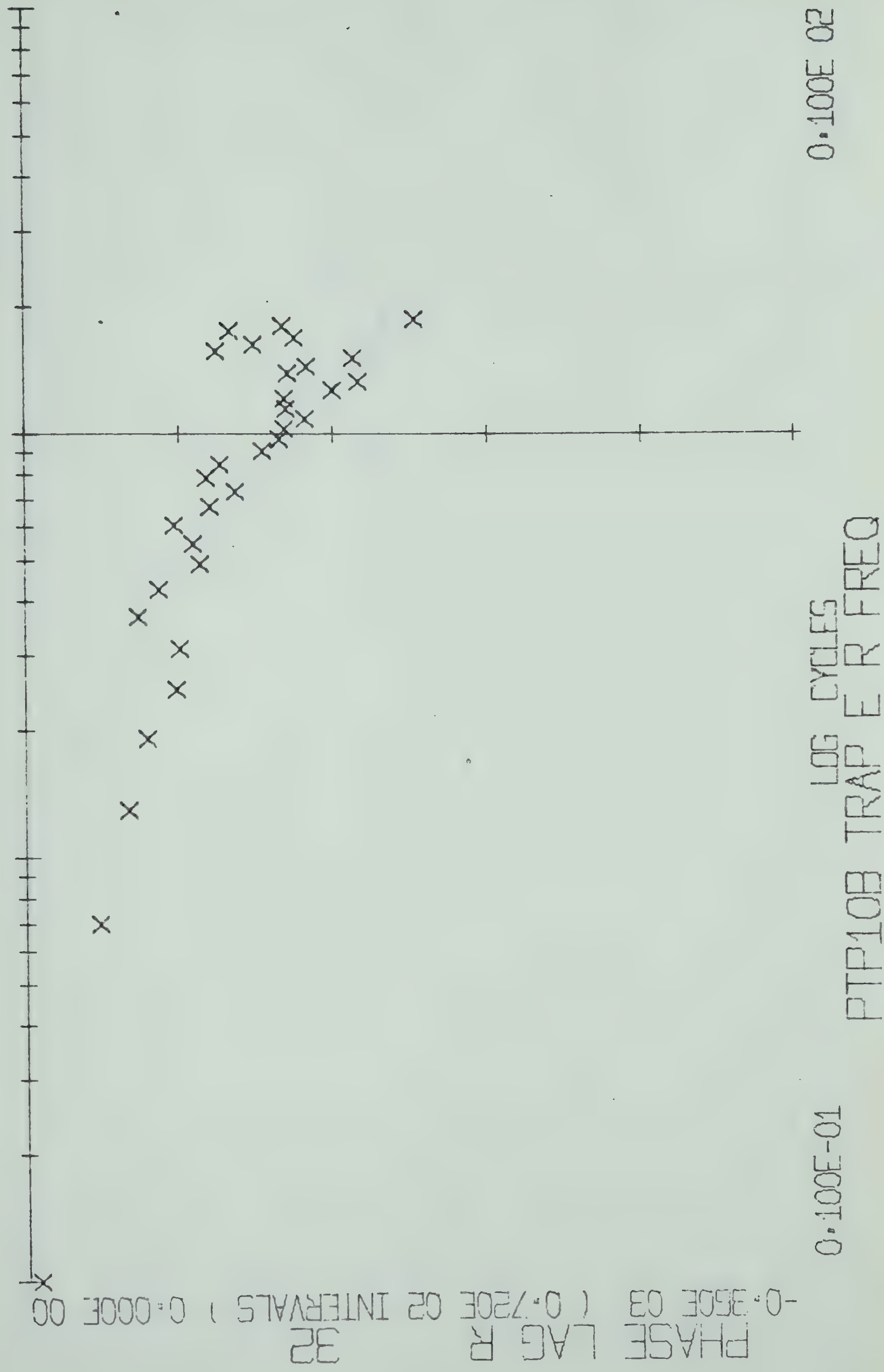

0.100E 02
0.100E-01
LOG CYCLES
PFT12A FAST E F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
15

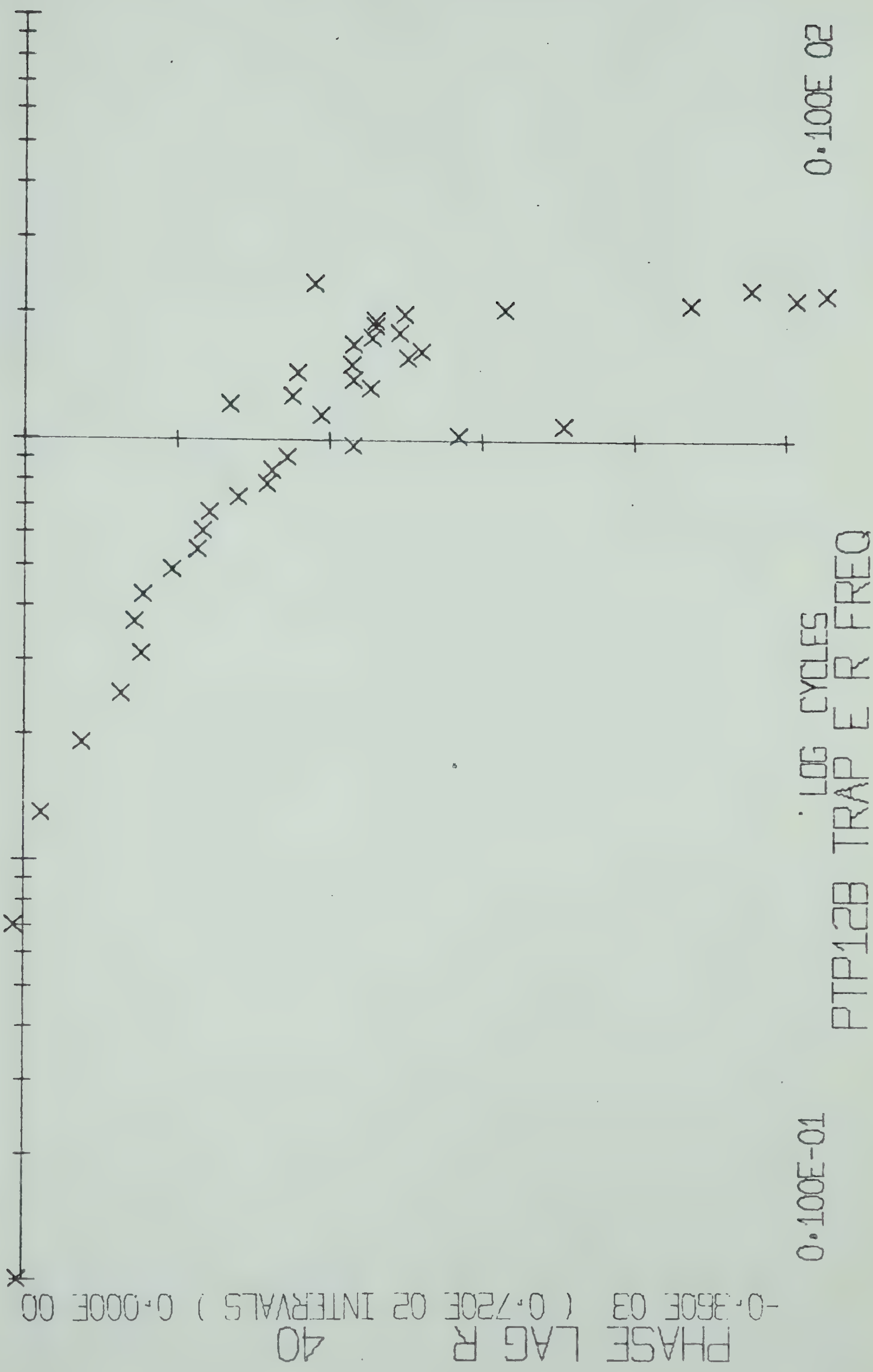

0.100E 02
LOG CYCLES
PFT16A FAST E F FREQ
0.100E-01
AMP. RATIO R 15
0.100E-01 LOG CYCLES
0.100E 01

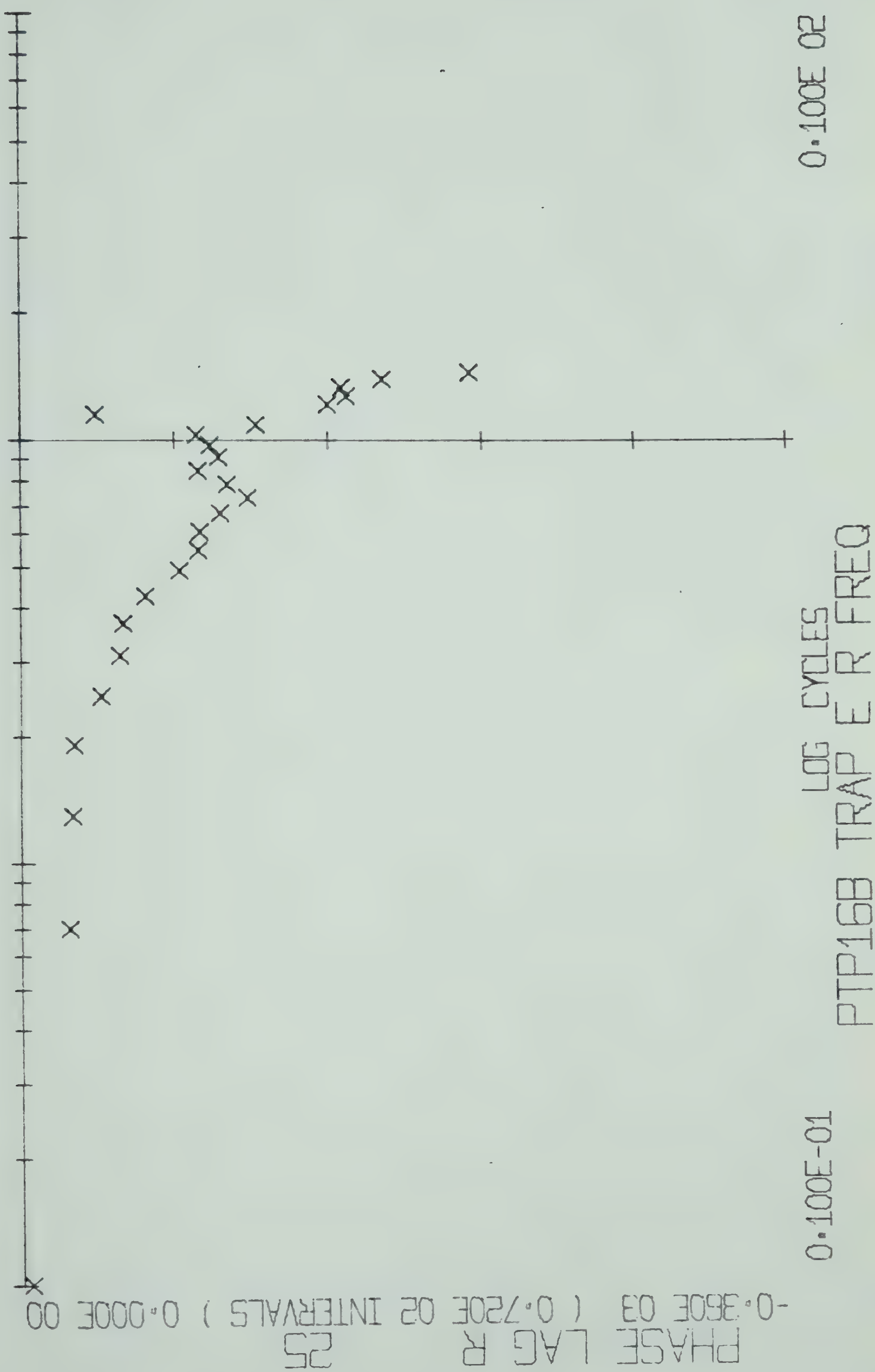

0.100E 02
LOG CYCLES
PFT17A FAST E F FREQ
0.100E-01
0.100E 01
0.100E-01 LOG CYCLES
AMP. RATIO R 15

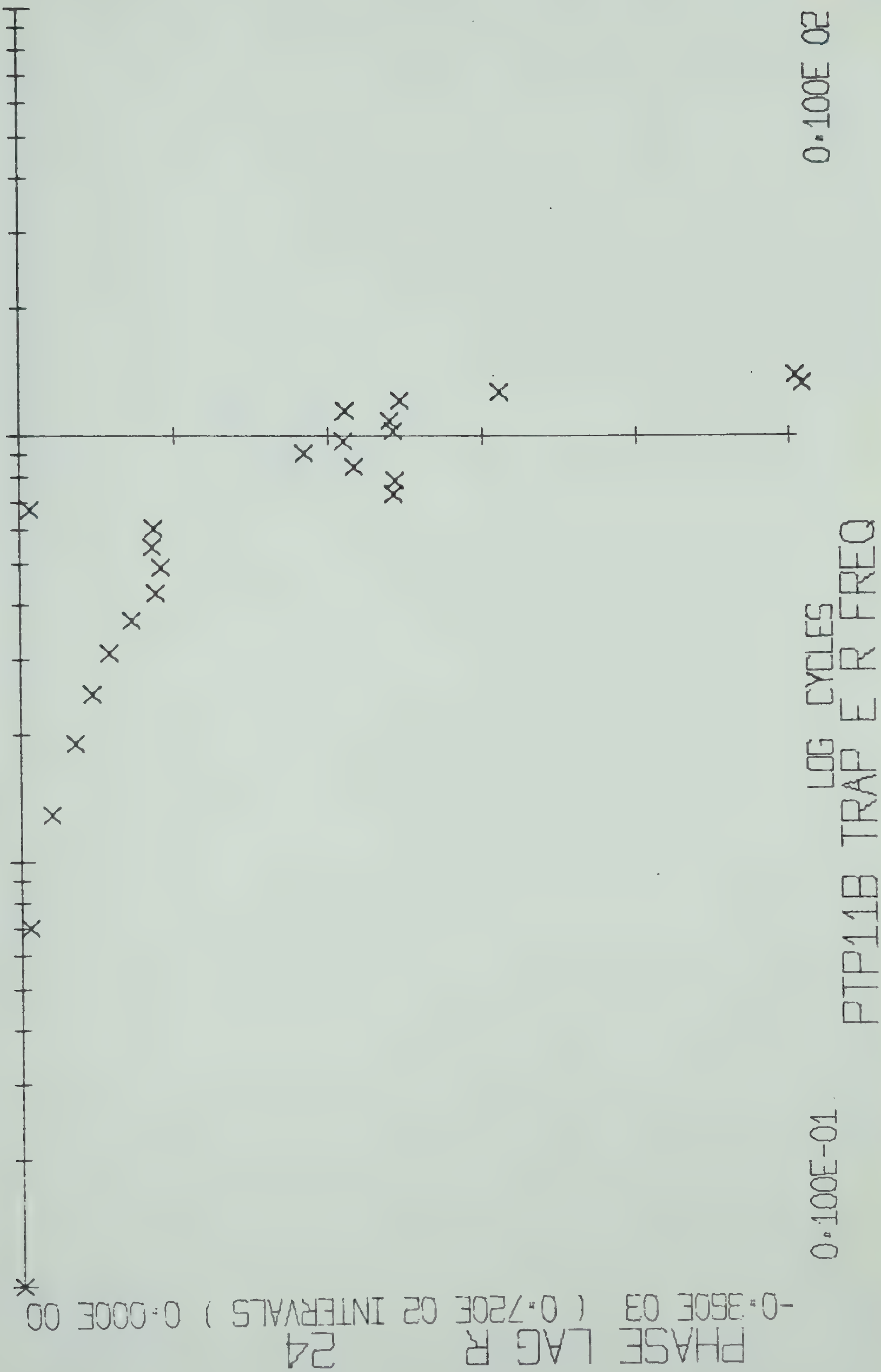

0.100E 02
0.100E-01
LOG CYCLES
PFT13A FAST E F FREQ
0.100E 01
0.100E-01
LOG CYCLES
AMP. RATIO R
15


0.100E 02
0.100E-01
LOG CYCLES
PTP14B TRAP E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
25

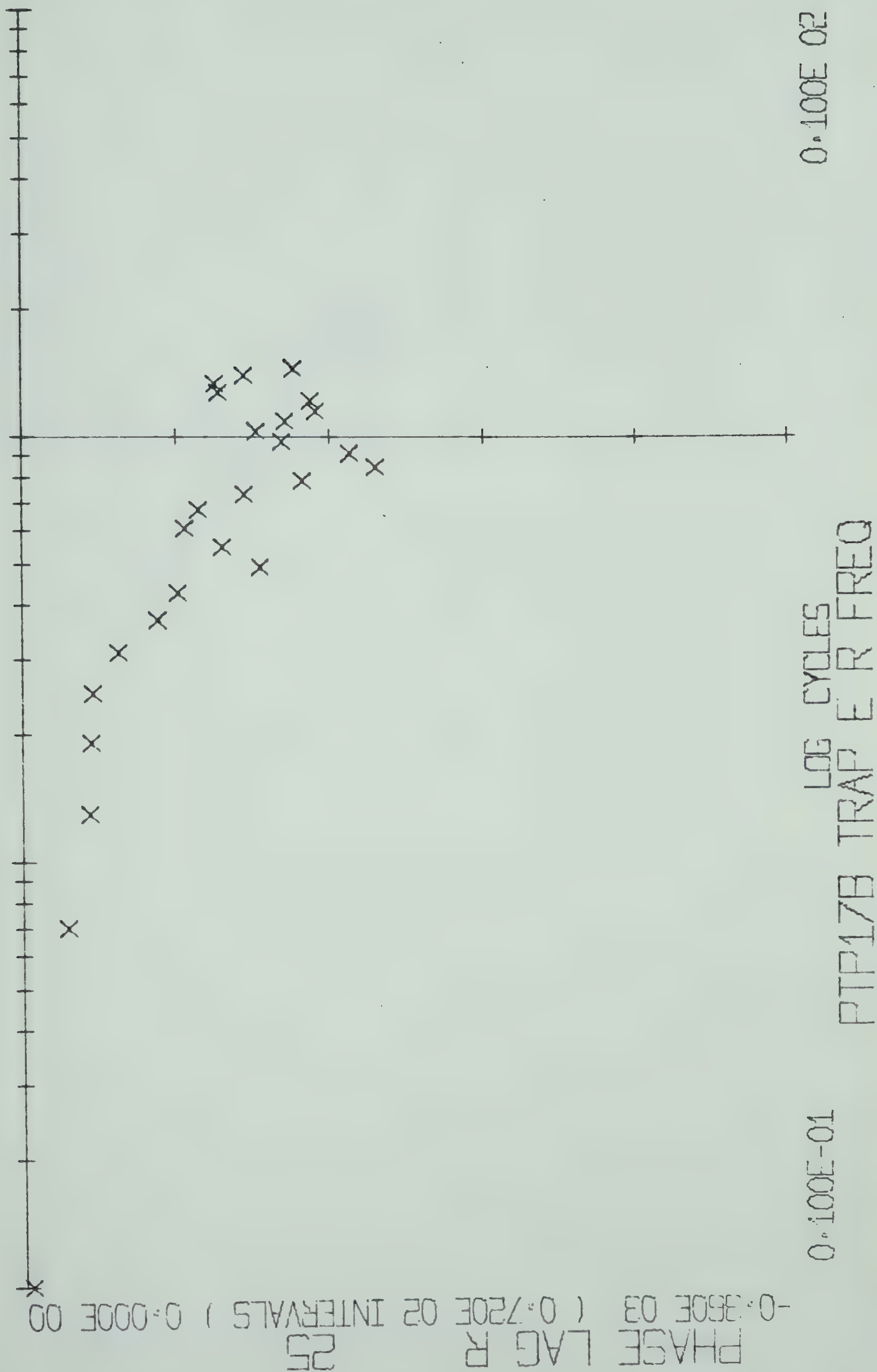

0.100E 02
0.100E-01
LOG CYCLES
PTP10B TRAP E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
32

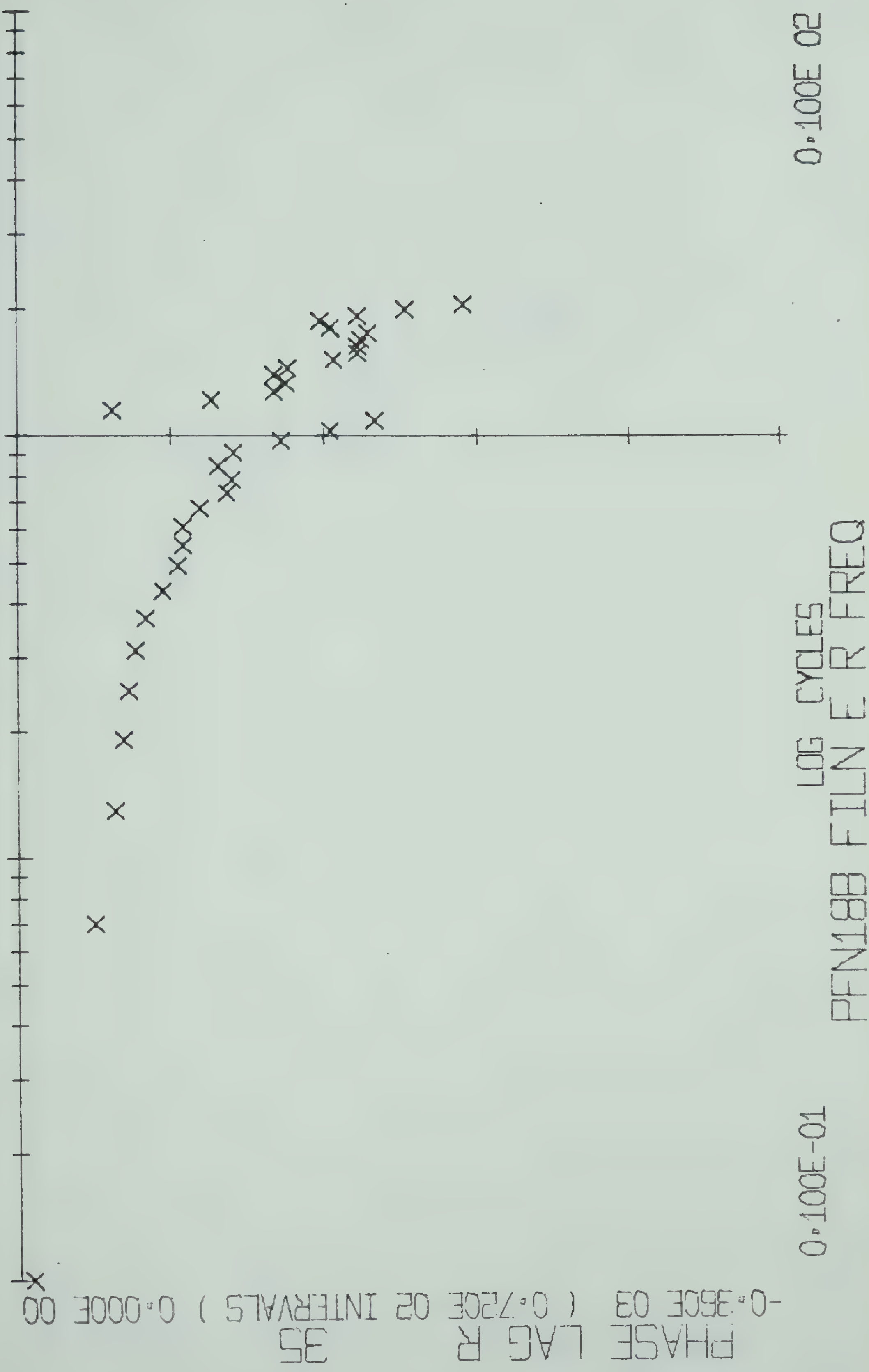

0.100E-01
0.100E 02
LOG CYCLES
PTP12B TRAP E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R 40

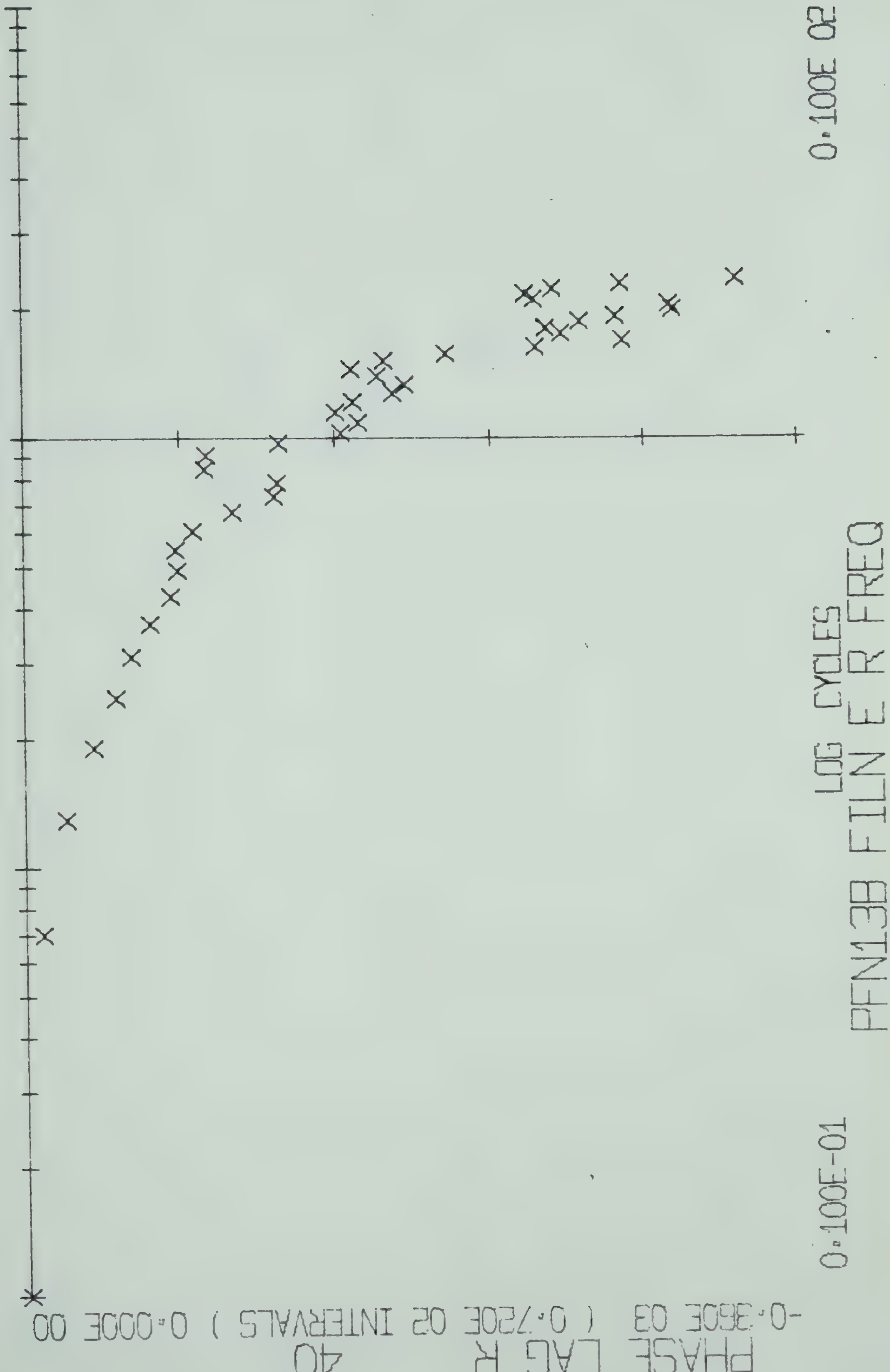

0.100E-01
0.100E 02
LOG CYCLES
PTP16B TRAP E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R 25

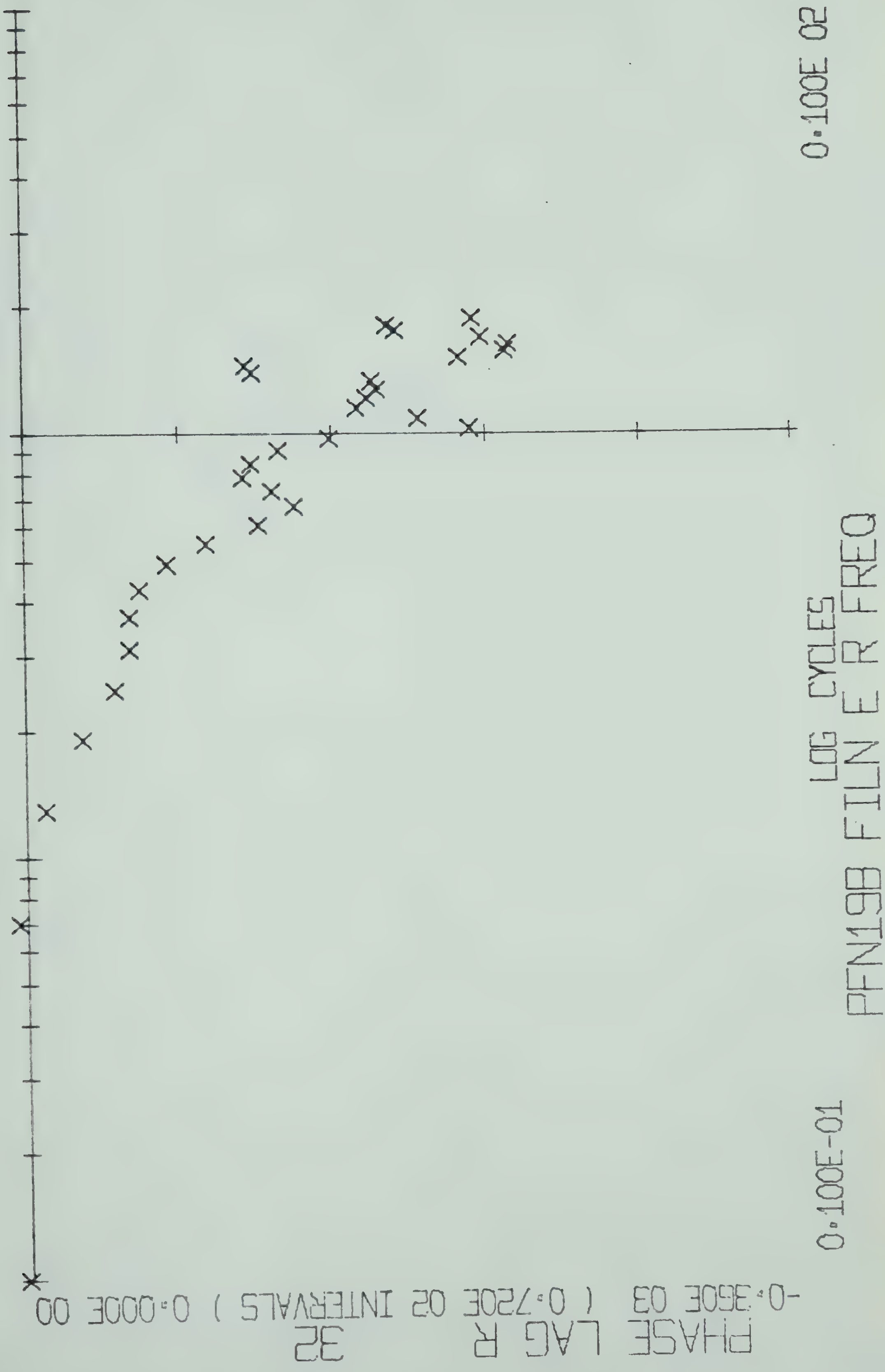


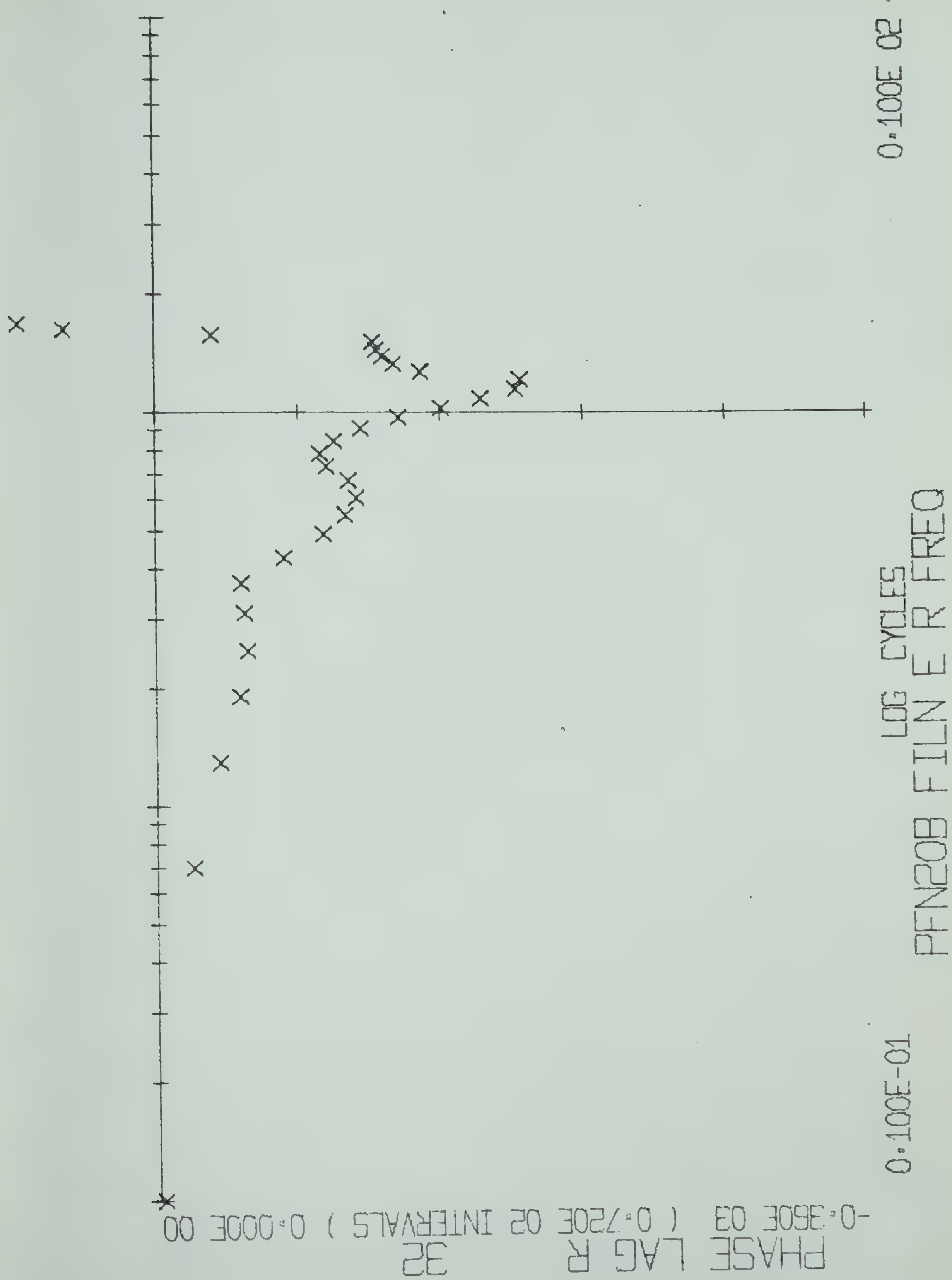
0.100E 02
LOG CYCLES
PTP11B TRAP E R FREQ
0.100E-01
PHASE LAG R 24
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00


0.100E 02
0.100E-01
LOG CYCLES
PTP17B TRAP E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R 25

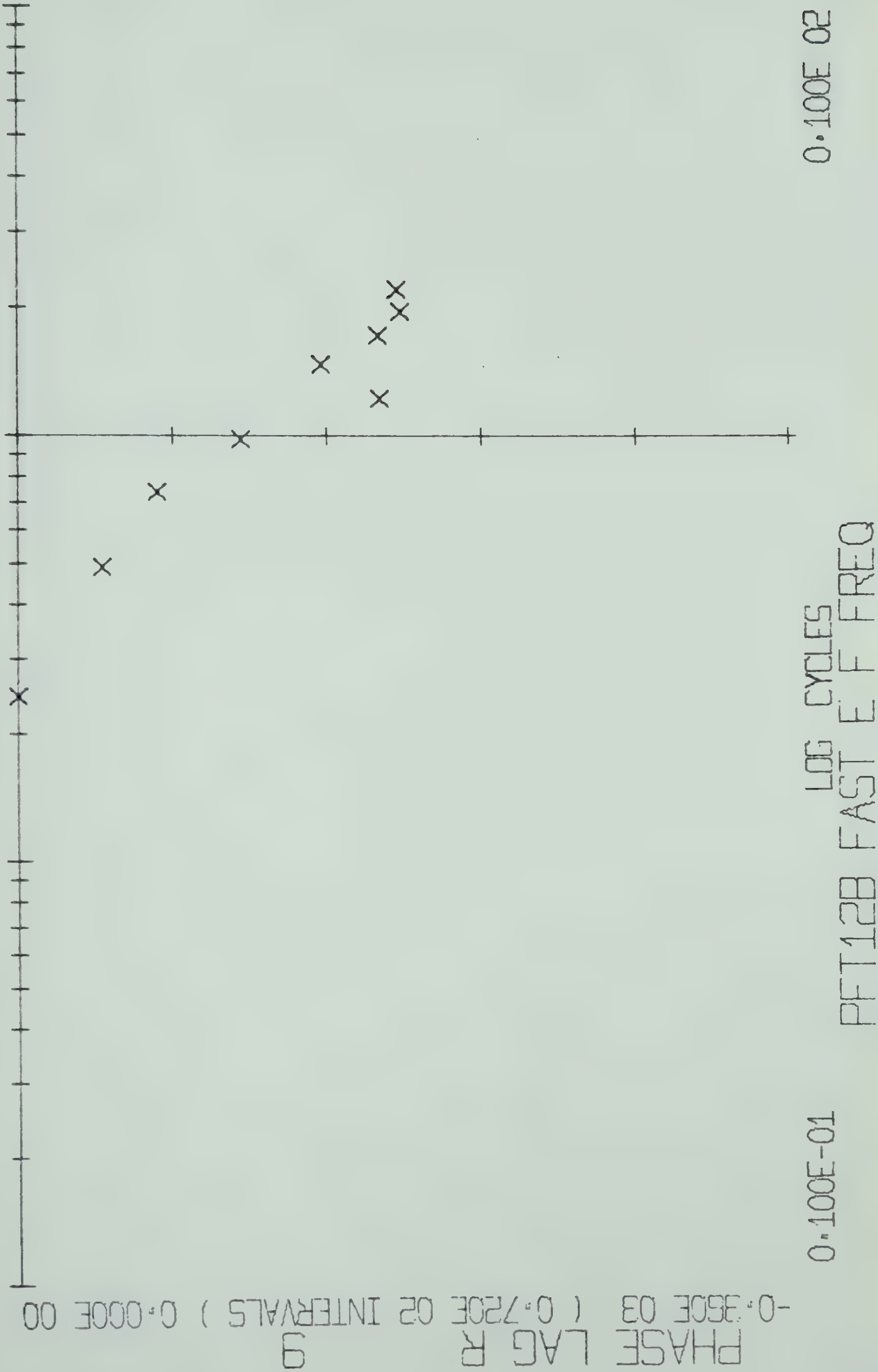

0.100E 02
0.100E-01
LOG CYCLES
PFN18B FILN E R FREQ
PHASE LAG R 35
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00

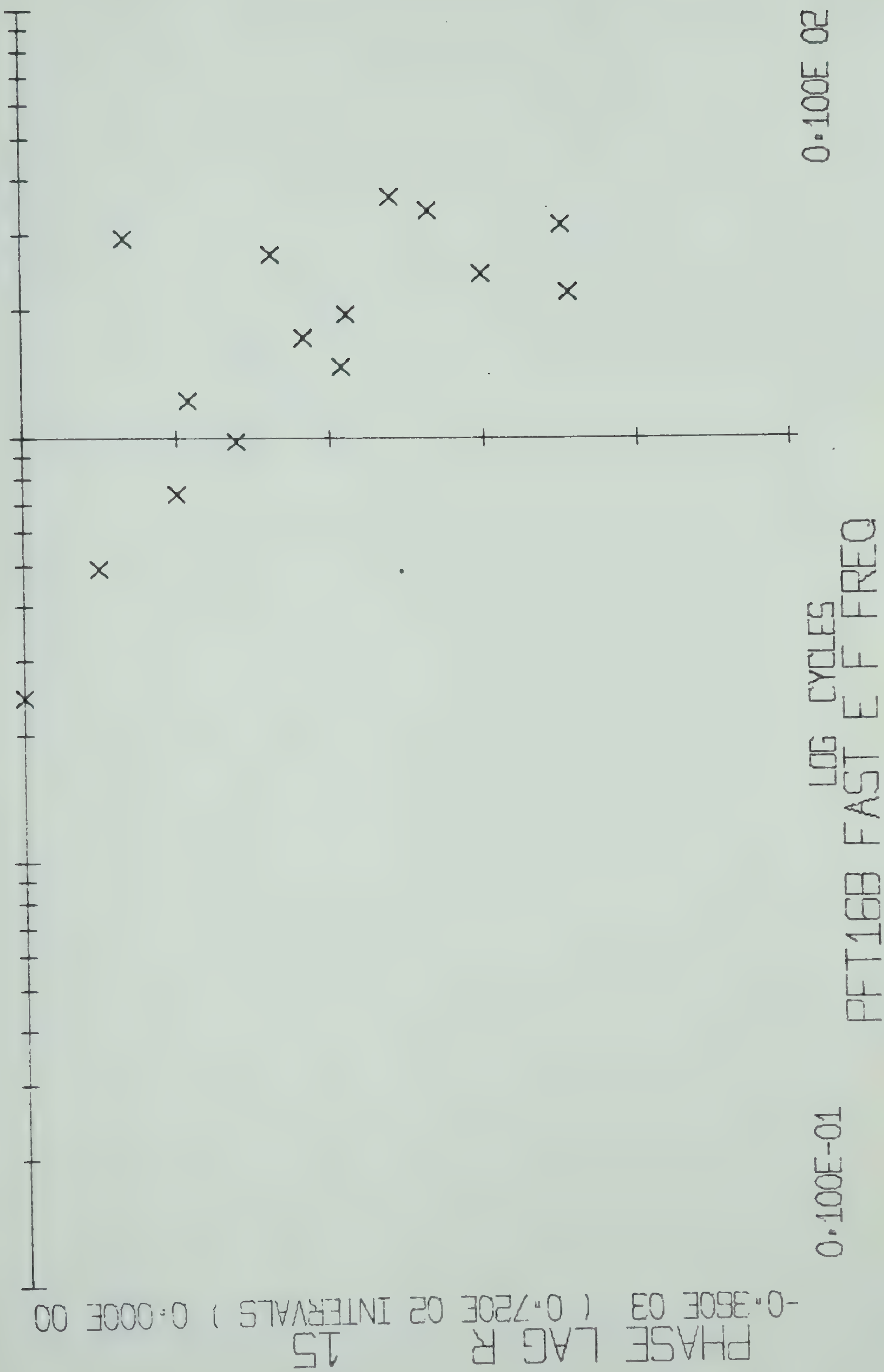

0.100E 02
0.100E-01
LOG CYCLES
PFN13B FILN E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
40


0.100E 02
0.100E-01
LOG CYCLES
PFN19B FILN E R FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
32

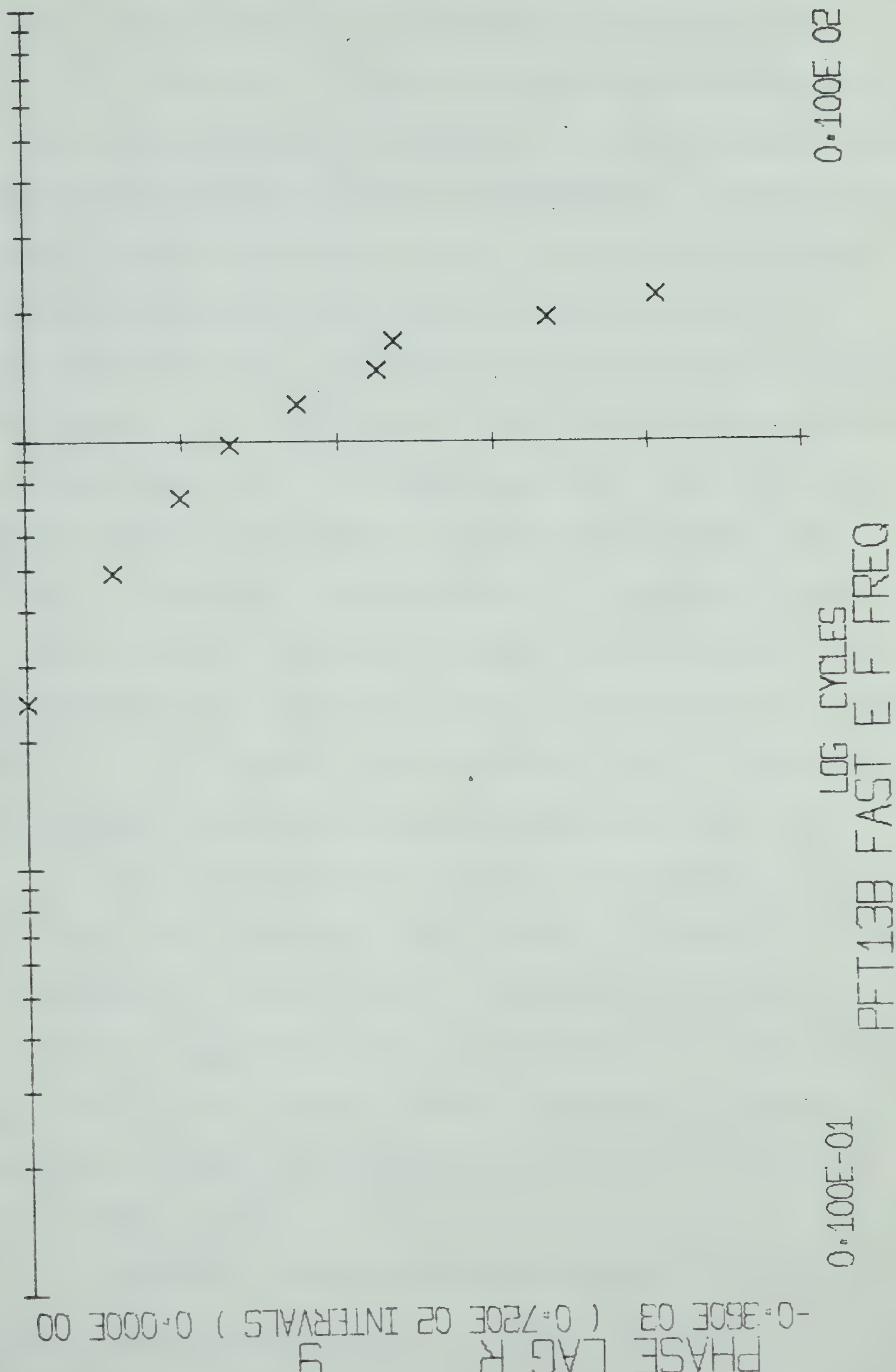

0.100E 02
0.100E-01
LOG CYCLES
PFN20B FILN E R FREQ
PHASE LAG R
32
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00

PHASE LAG R
6
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
LOG CYCLES
PFT14B FAST E F FREQ
0.100E-01
0.100E 02


0.100E 02
0.100E-01
LOG CYCLES
PFT12B FAST E F FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
9


0.100E 02
0.100E-01
LOG CYCLES
PFT16B FAST E F FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
15


0.100E 02
0.100E-01
LOG CYCLES
PFT17B FAST E F FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
15


0.100E 02
0.100E-01
LOG CYCLES
PFT13B FAST E F FREQ
-0.360E 03 (0.720E 02 INTERVALS) 0.000E 00
PHASE LAG R
9

The time smoothing involved applying a third order polynomial least square fit using eleven data points as discussed in Section 5.4.2.2. The effect of this smoothing on the time data can be seen by comparing various graphs in Appendix II. The data used to plot Graph PEX13A when smoothed is plotted in Graph PEX13I. Similarly, the data plotted in Graph PEX13B when smoothed, appears as shown by Graph PEX13J. It was apparent from these graphs that the smoothing was successful in removing some of the higher frequency noise on the signal, but it was unsuccessful in altering the large discontinuities introduced by the DDC flow control on a one second scan cycle. The effect of the high-frequency noise on the frequency response data appeared to be insignificant. For example, in Section 6.3.3.1, comparison of Graph PFT13E, which is obtained from raw time data, and Graph PFT13W, obtained from smoothed time data, demonstrates that the effect of time smoothing negligible. It is the author's opinion that the effect of the error of closure overshadows any effect the high-frequency noise might have.

The spectral smoothing was applied to experimental results with and without time smoothing. In all cases, as the graphs in Section 6.3.3.1 demonstrate, spectral smoothing did nothing to improve results and in some cases, appeared to cause deterioration, i.e., Graph PFT13H. It should also be noted, however, that the application of spectral smoothing in the presence of an error of closure did not improve the results in Section 6.2.4.

A study of the frequency content graphs in Section 6.3.3.1 reveals that spectral smoothing did not tend to remove the zeros of the

frequency content curves as it had for the theoretical curves of Section 6.2.3.1. In some cases, the spectral smoothing tended to introduce "false" zeros in the frequency content curves from experimental pulses. This may account for the fact that the results presented in Graph PFT13H appear to have deteriorated as a consequence of spectral smoothing.

6.3.3.1 Amplitude Ratio and Frequency Content Graphs Illustrating the Effect of Time and Spectral Smoothing

General comments:

a. Table 9 describes the graphs presented on pages 244 - 257 in the order in which they appear.

b. A description of the code used for labelling the axes of these graphs was given in Section 6.1.

c. The data presented in this section was derived from the time domain pulse data for run 13 which is presented in Appendix II. The data presented in this section was derived from the time domain pulse data for run 13 which is presented in Appendix II. The Fast Fourier transform was used to calculate the results for all cases except Graph PFN13W which was derived using Filon's quadrature.

d. All amplitude ratio graphs except Graph PFN13W are on the same scale to facilitate comparisons.

e. All the frequency content graphs have the same abscissa scale but not the same ordinate scale.

f. Each of the seven amplitude ratio graphs presented on pages 244 - 251 represent the results of following one of the four possible paths in the structure below from raw time data to amplitude ratio curve:

$$\begin{tikzcd}
\text{raw time data} \arrow[r] \arrow[d] & \text{time smoothing} \arrow[dl] \\
\text{Fourier transform} \arrow[r] \arrow[d] & \text{spectral smoothing} \arrow[dl] \\
\text{amplitude ratio} &
\end{tikzcd}$$

g. The error in the smoothing program which was discussed in Section 6.2.3 and demonstrated in the graphs of Section 6.2.3.1 was present when the smoothed graphs presented in this Section were obtained. However, the effect of the error was minor so the qualitative effect of smoothing observed in the graphs of this section can be considered valid.

h. For information on the sample interval used on the time data associated with these runs, see item g, Section 6.3.1.2.

6.4 Conclusions

In general, pulse testing proved to be a useful tool for dynamic testing. It should not be too difficult to obtain valid frequency response data from most chemical processes with a reasonable grasp of the mathematical theory of pulse testing and the experience of a few tests conducted on a known simple process, such as a first order system. In the author's opinion, a successful pulse test requires a certain amount of intuitive judgment with respect to the choice of the input pulse shape amplitude and duration. There does not appear to be a "best" pulse for all situations, and to the best of the author's knowledge, there are no general guidelines for pulse selection published in the literature. It appears that the best policy is to try several different pulses on each process until experience allows elimination of certain ones.

TABLE 9

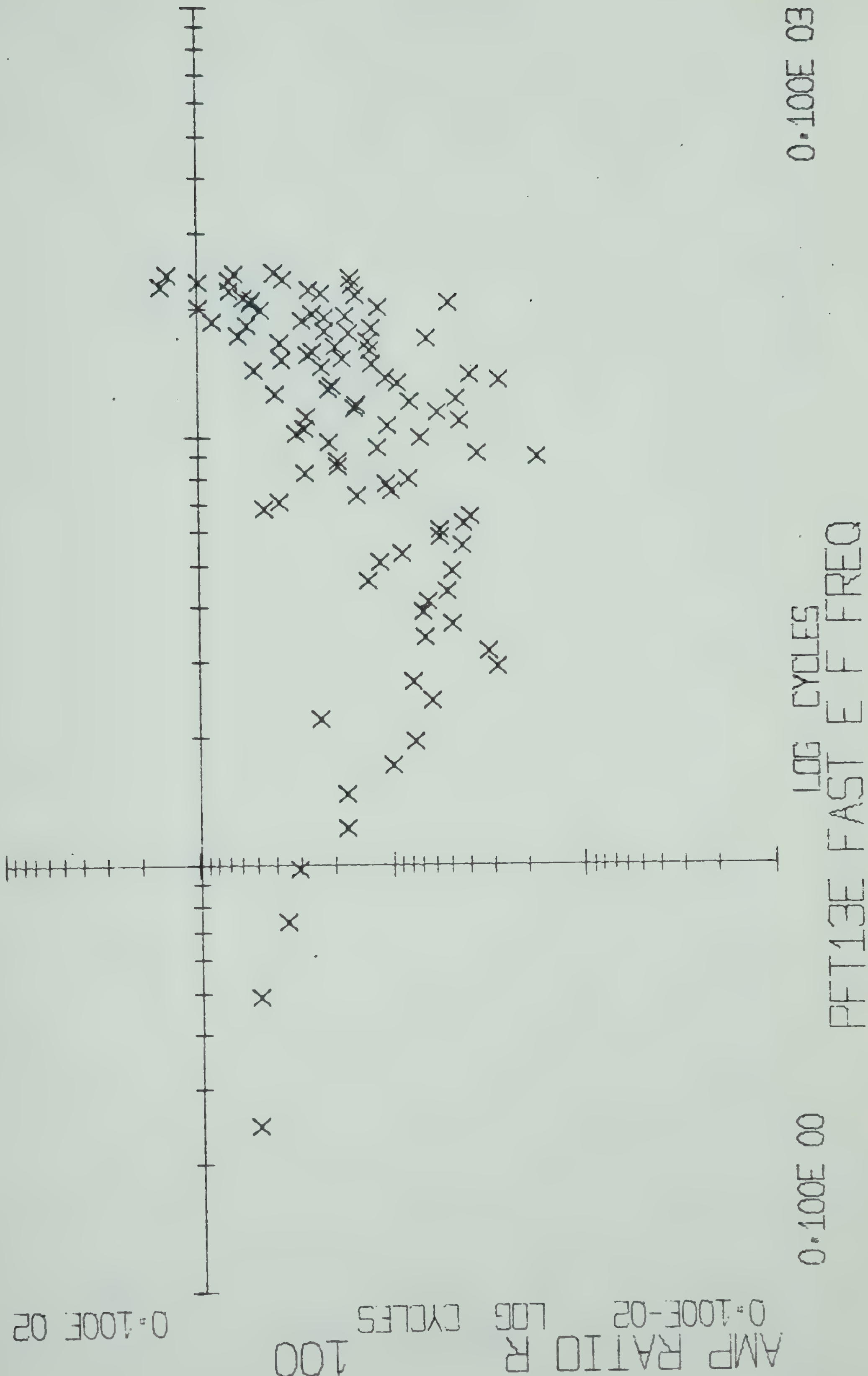
LIST OF AMPLITUDE RATIO AND FREQUENCY CONTENT GRAPHS ILLUSTRATING THE EFFECT OF SMOOTHING ON EXPERIMENTAL RESULTS

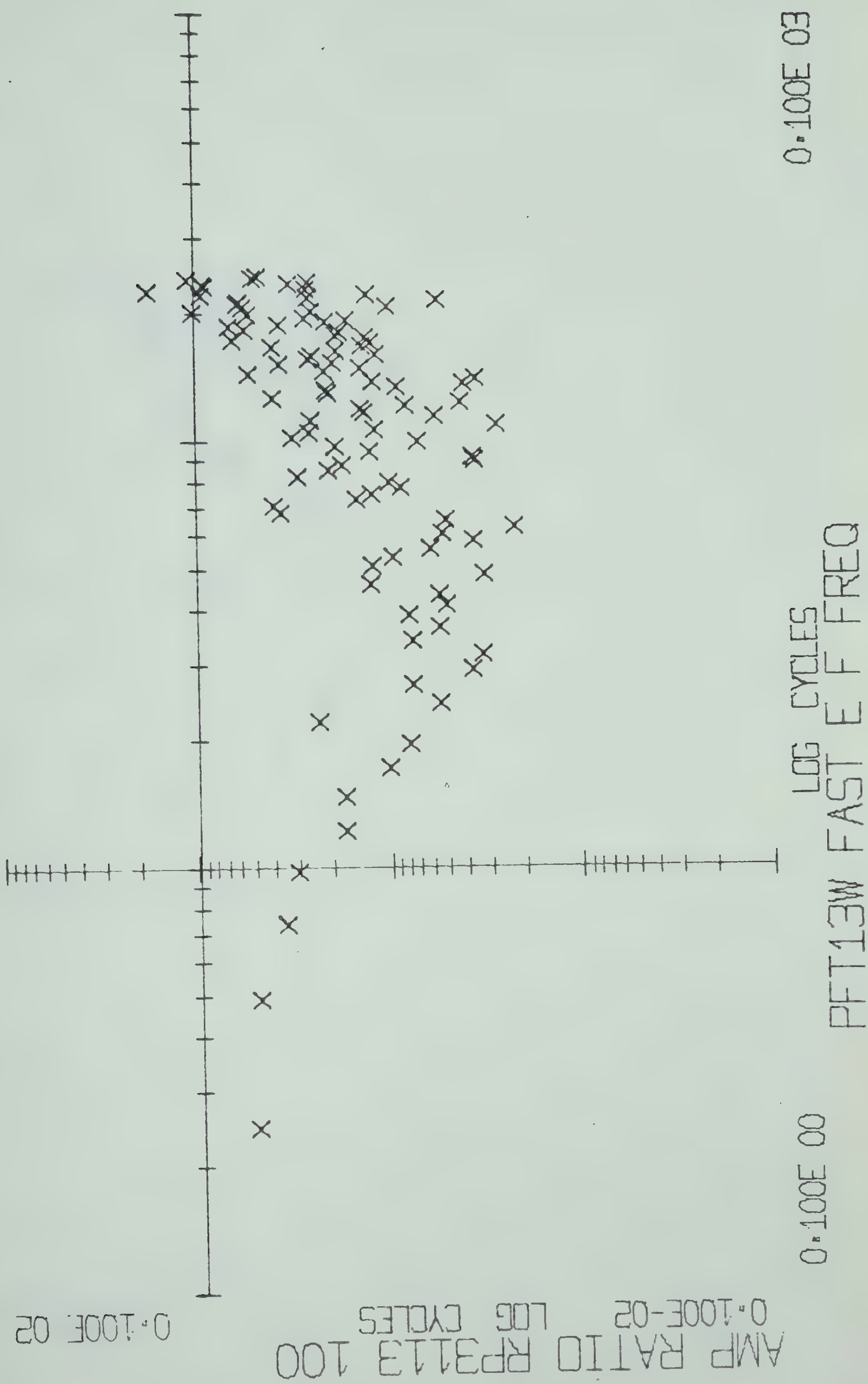
GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFT13E	244	FT	Amplitude ratio; no smoothing.
PFT13W	245	FT	Amplitude ratio; time smoothing using a third order polynomial eleven point smooth applied three times.
PFT13U	246	FT	Amplitude ratio; spectral smoothing using Hamming method.
PFT13N	247	FN	Amplitude ratio; time smoothing using a third order polynomial eleven point smooth applied once and spectral smoothing using Hamming method.
PFT13Y	248	FT	Amplitude ratio; data smoothing the same as for Graph PFT13N, except time smoothing is applied three times.
PFT13H	249	FT	Amplitude ratio; spectral smoothing using Hanning method.
PFT13D	250	FT	Amplitude ratio; time smoothing using a third order polynomial eleven point smooth applied three times, and spectral smoothing using Hanning method.

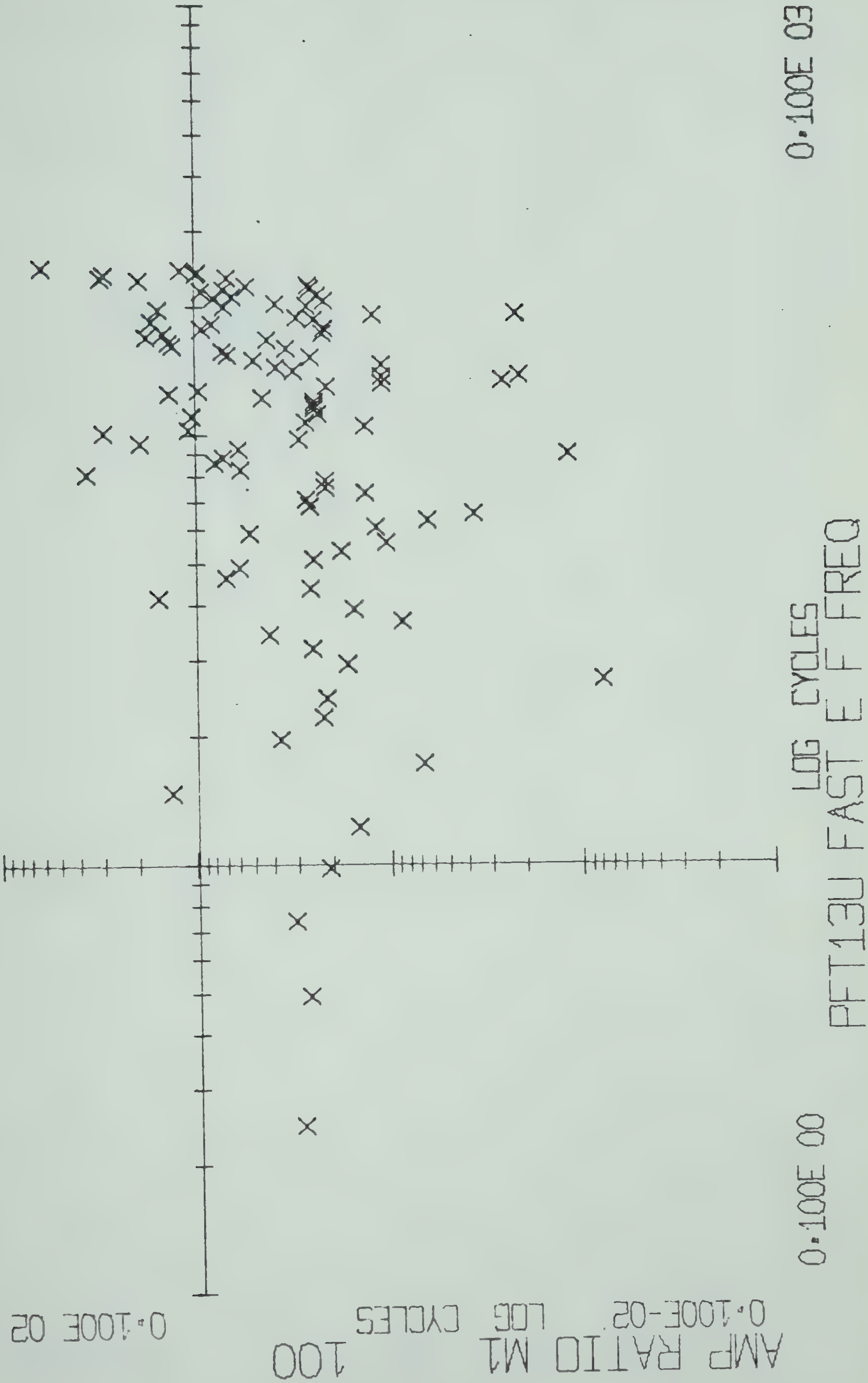
TABLE 9 (CONT'D)

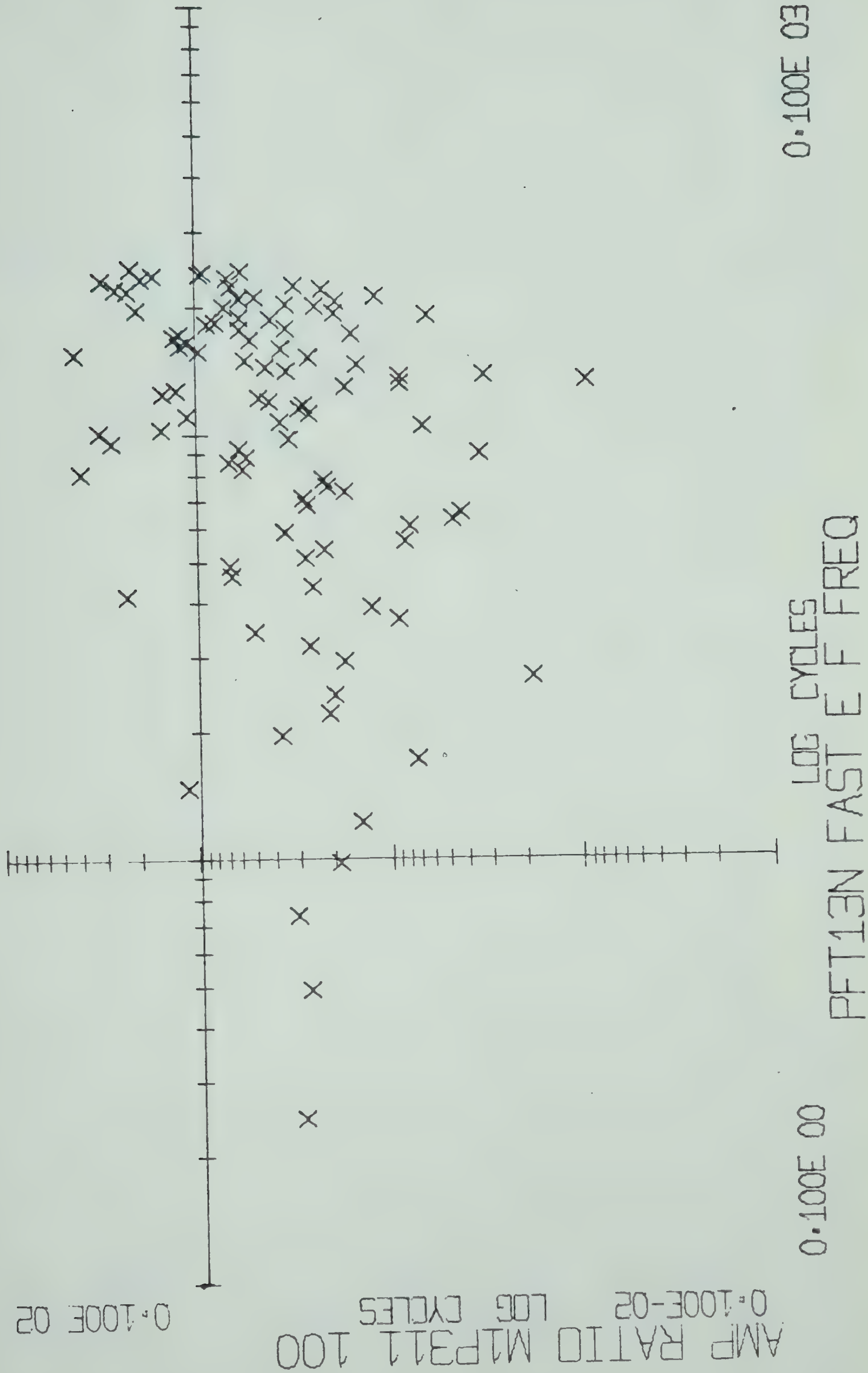
LIST OF AMPLITUDE RATIO AND FREQUENCY CONTENT GRAPHS
ILLUSTRATING THE EFFECT OF SMOOTHING ON EXPERIMENTAL RESULTS

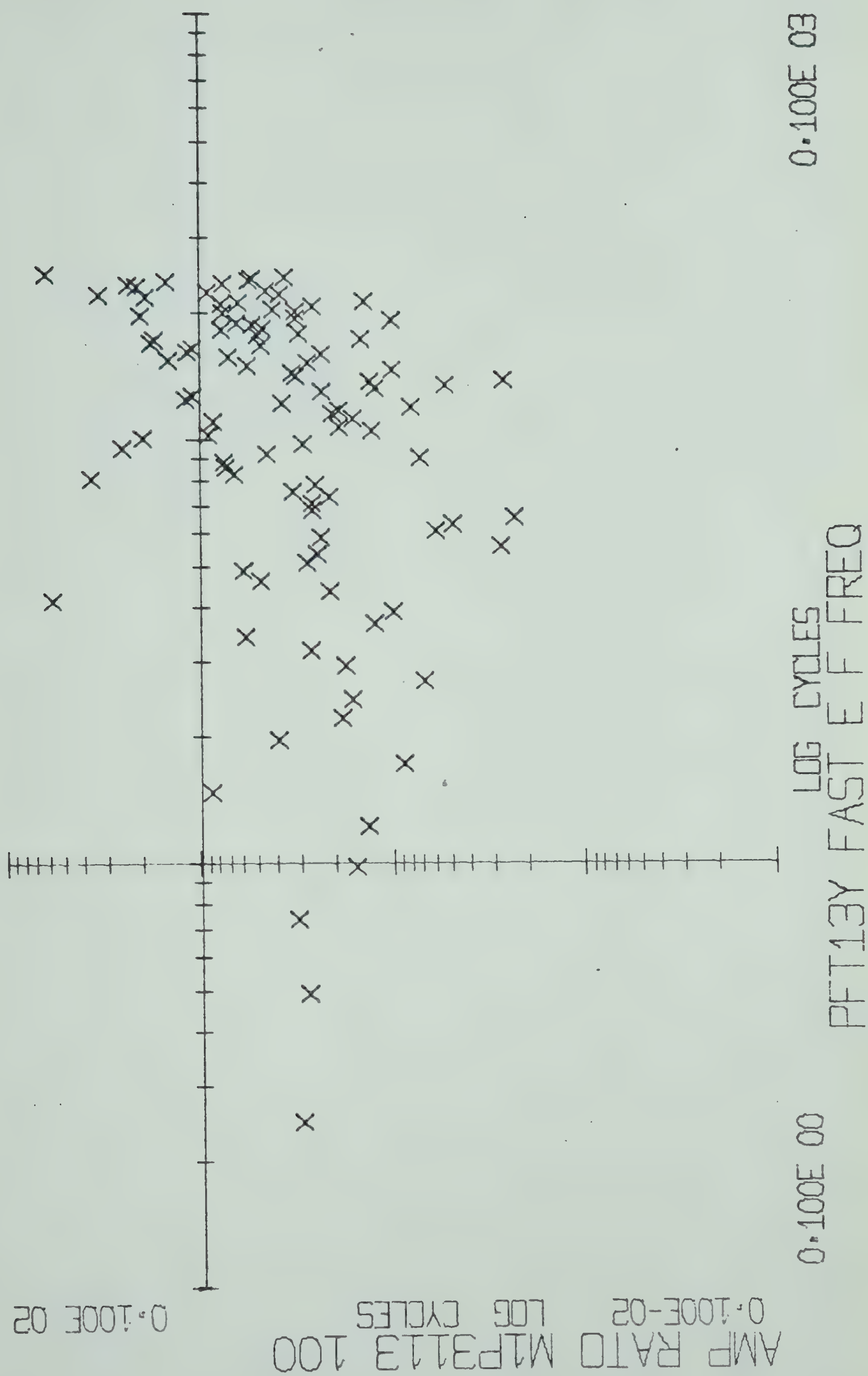
GRAPH IDENTIFICATION	PAGE	FOURIER TRANSFORM TECHNIQUE	COMMENT
PFN13W	251	FN	Amplitude ratio; time smoothing using a third order polynomial eleven point smooth applied three times.
PFT13G	252	FT	Frequency content; no smoothing.
PFT13X	253	FT	Frequency content; data smoothing the same as for Graphs PFT13W and PFN13W.
PFT13P	254	FT	Frequency content; data smoothing the same as for Graph PFT13N.
PFT13Z	255	FT	Frequency content; data smoothing the same as for Graph PFT13Y.
PFT13J	256	FT	Frequency content; data smoothing the same as for Graph PFT13H.
PFT13Q	257	FT	Frequency content; data smoothing the same as for Graph PFT13D.

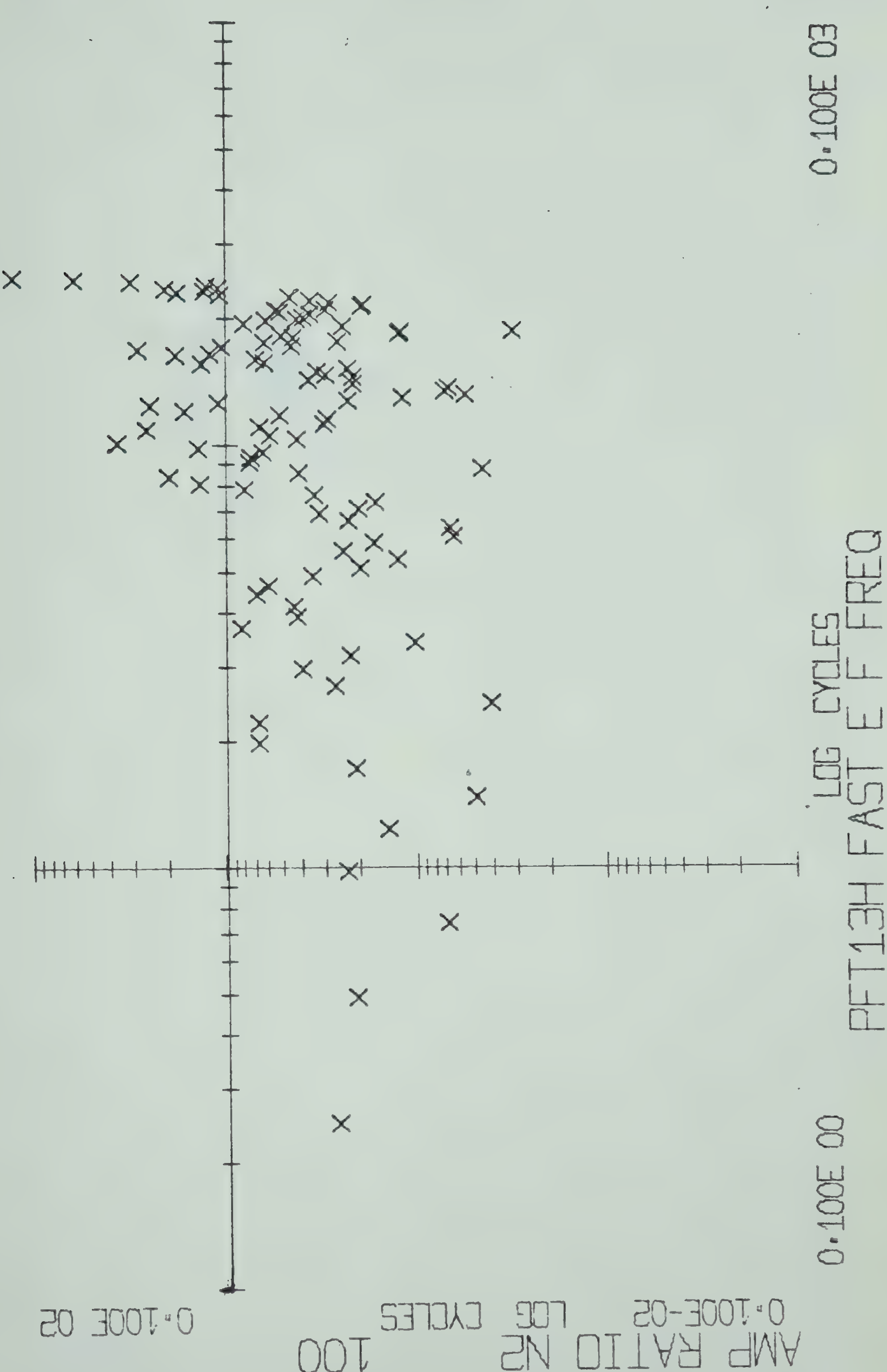

0.100E 03
0.100E 00
LOG CYCLES
PFT13E FAST E F FREQ
AMP RATIO R
100
0.100E-02
LOG CYCLES
0.100E 02

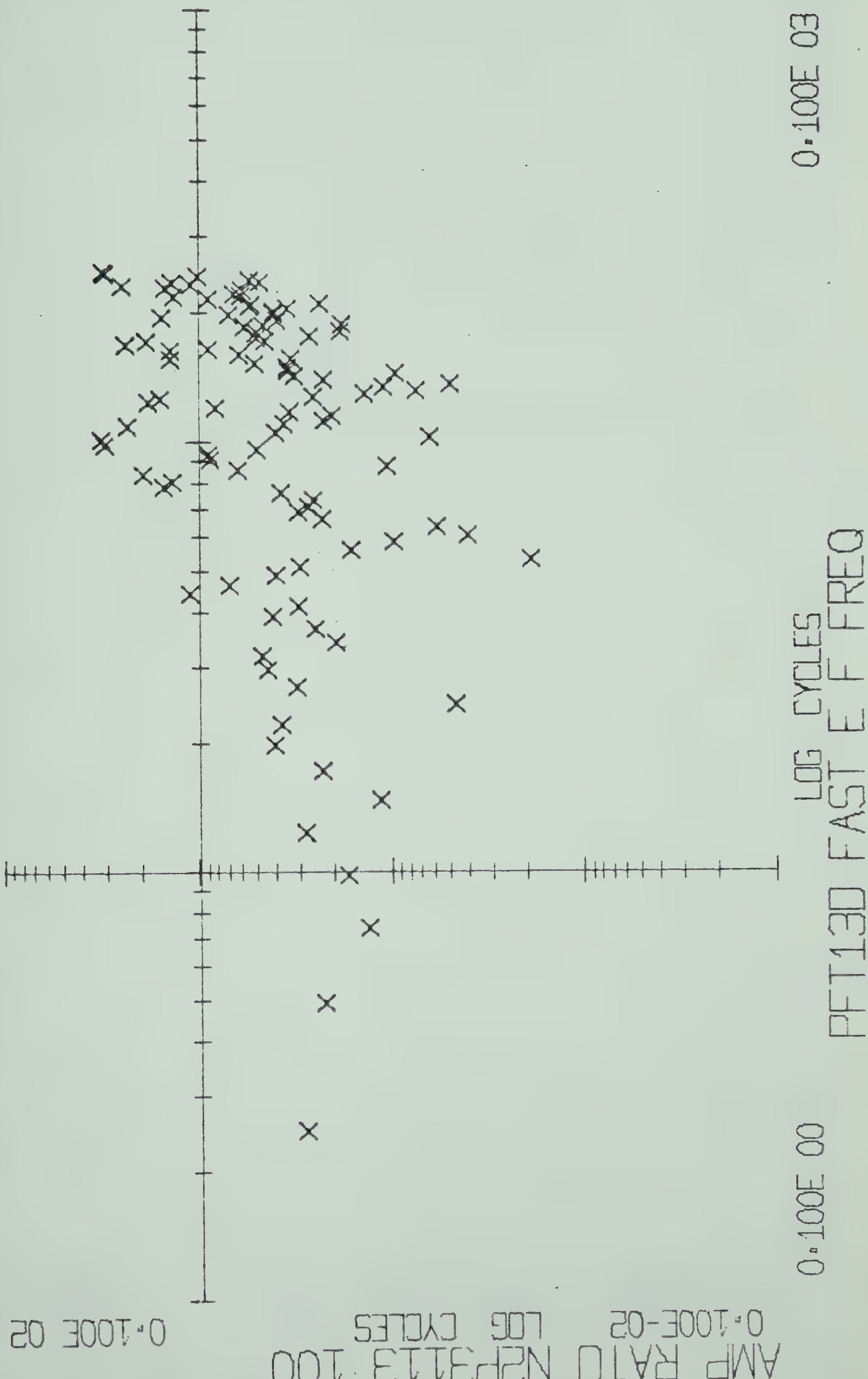

0.100E 03
LOG CYCLES
PFT13W FAST E F FREQ
0.100E 00
0.100E-02
LOG CYCLES
AMP RATIO RP3113 100
0.100E 02

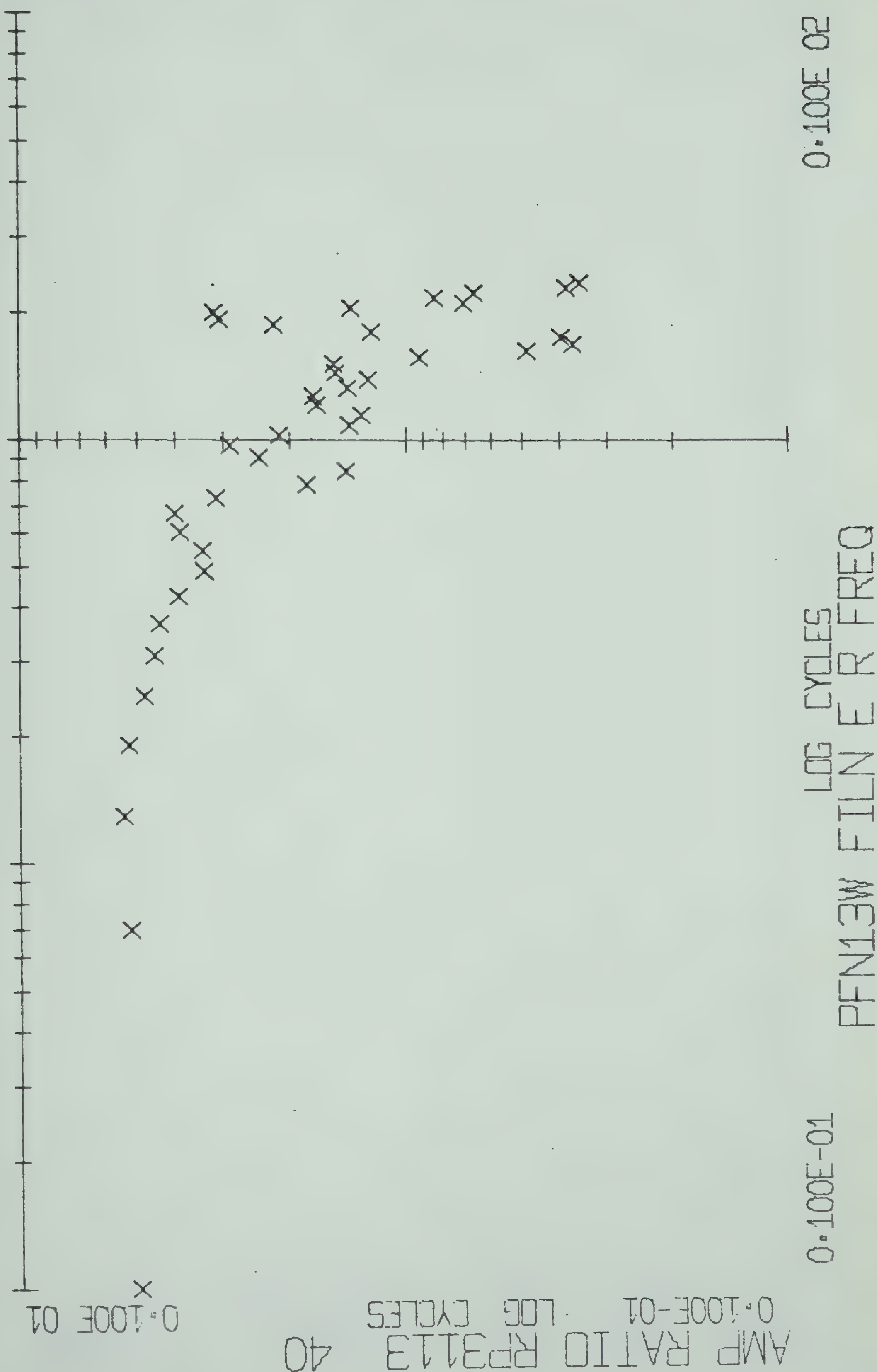

0.100E 03
LOG CYCLES
PFT13U FAST E F FREQ
0.100E 00
0.100E-02 LOG CYCLES
AMP RATIO M1
100
0.100E 02

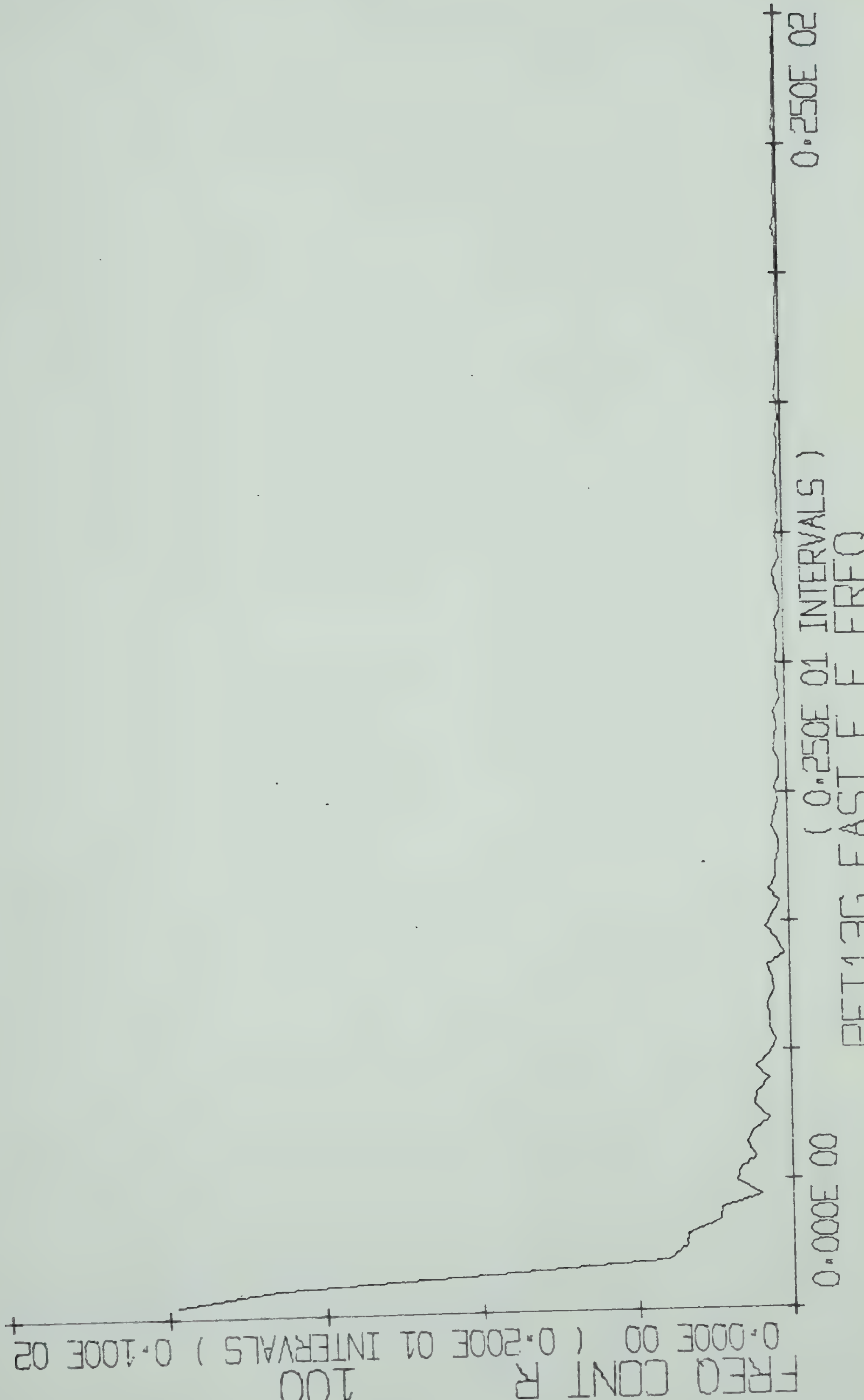

0.100E 03
LOG CYCLES
PFT13N FAST E F FREQ
0.100E 00
0.100E 02
0.100E-02 LOG CYCLES
AMP RATIO M1P311 100

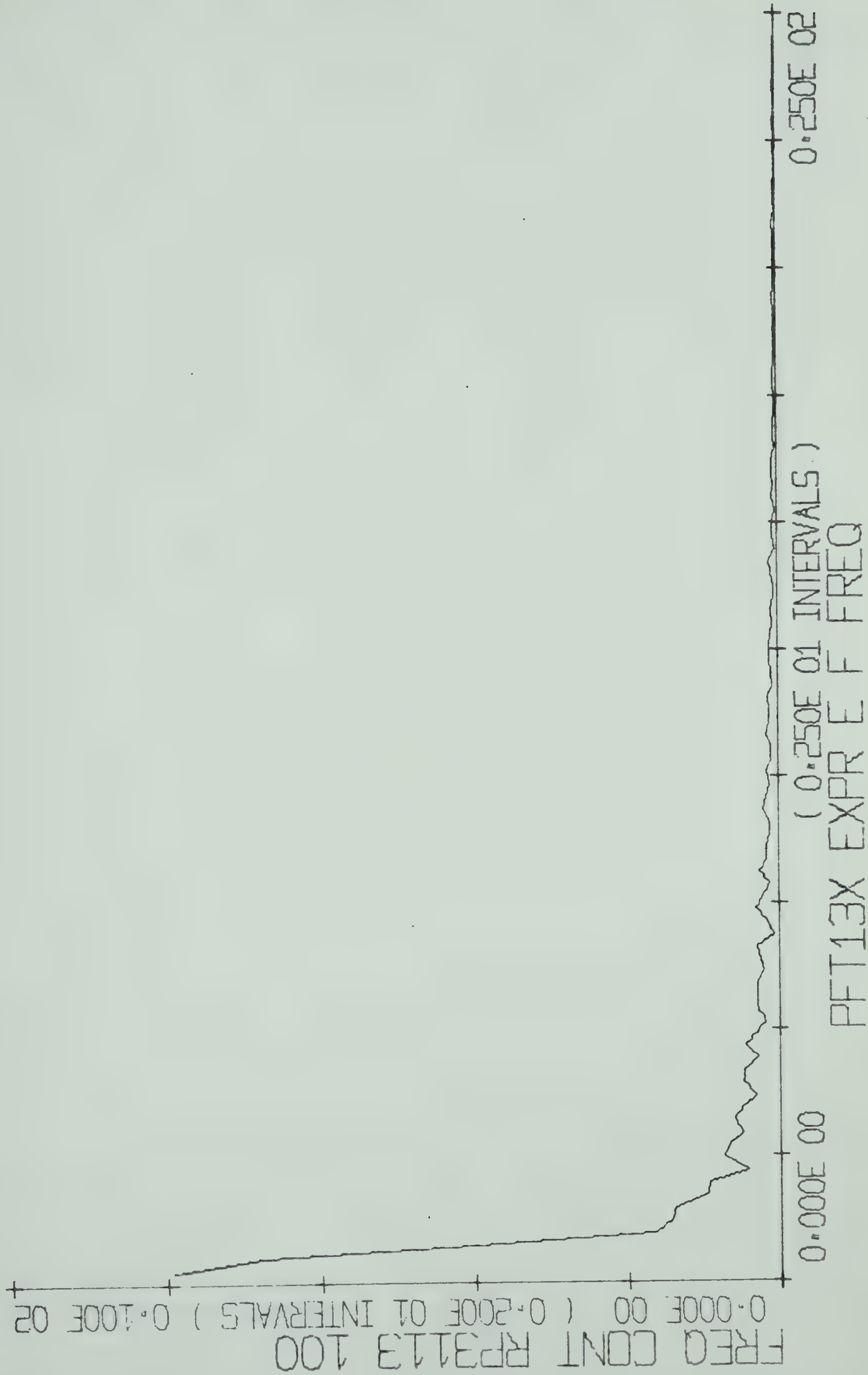

0.100E 03
LOG CYCLES
PFT13Y FAST E F FREQ
0.100E 00
0.100E 02
LOG CYCLES
0.100E-02
AMP RATIO M1P3113 100

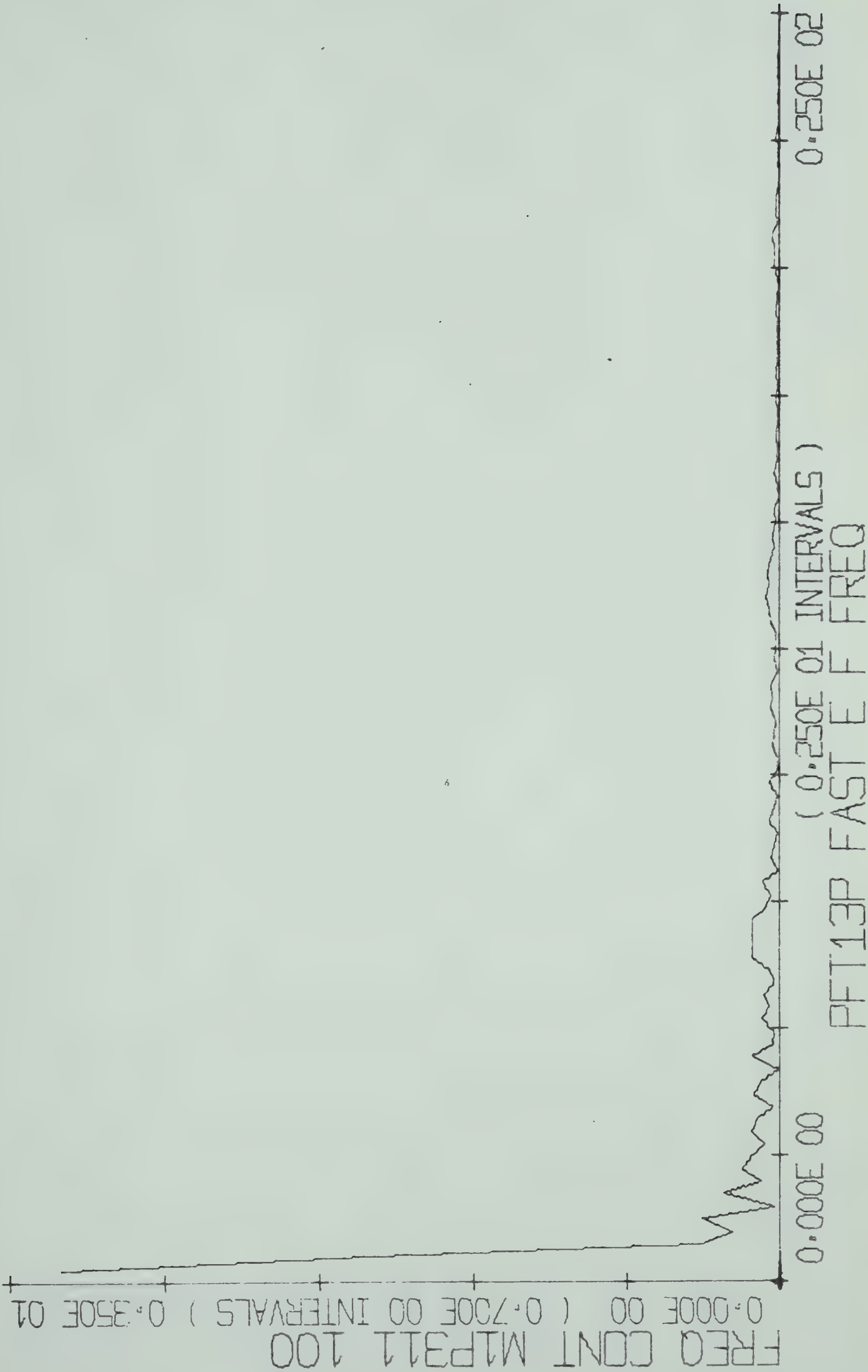

0.100E 03
LOG CYCLES
PFT13H FAST E F FREQ
0.100E 00
AMP RATIO N2
0.100E-02
LOG CYCLES
100
0.100E 02

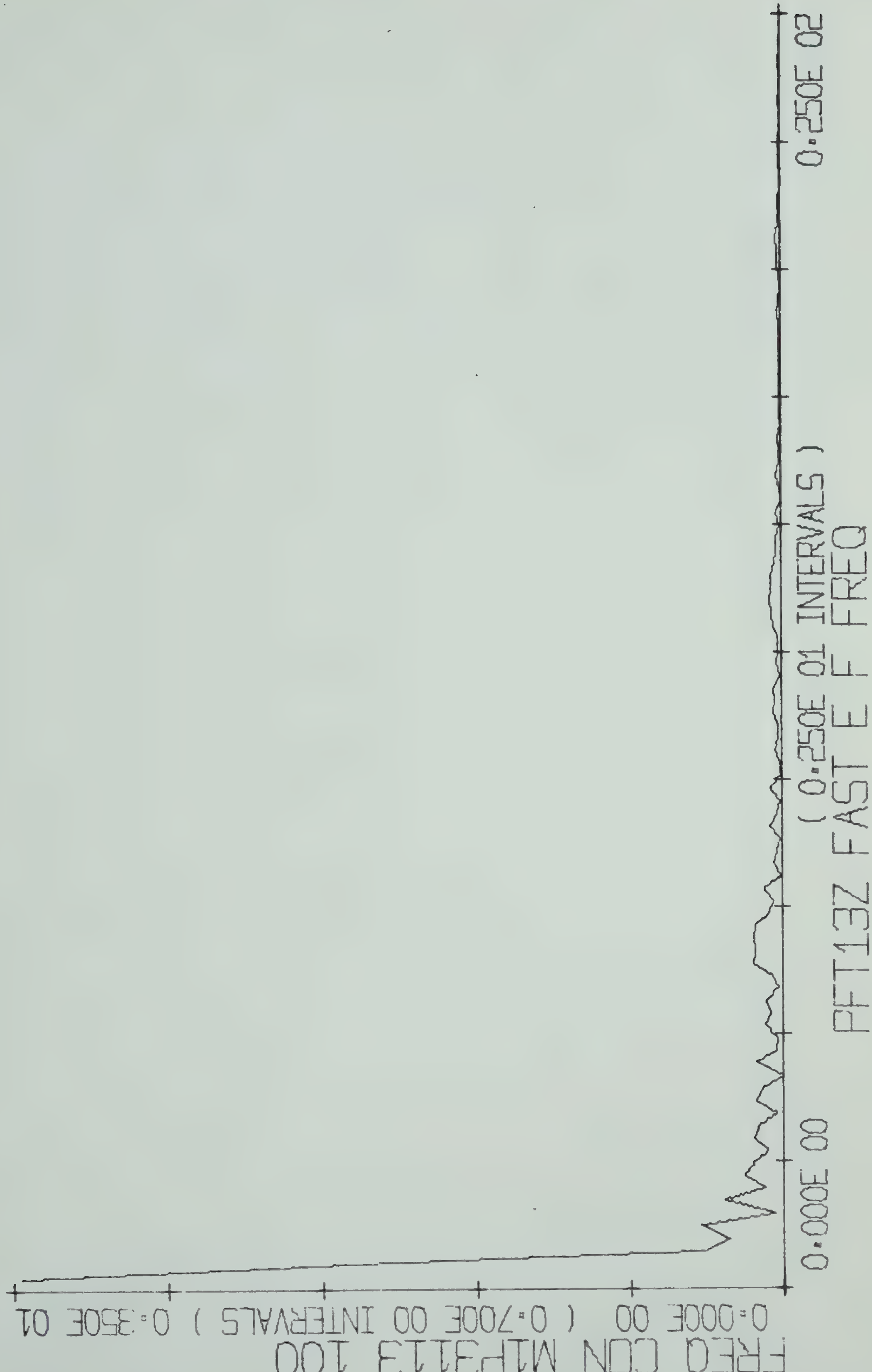

0.100E 03
LOG CYCLES
PFT13D FAST E F FREQ
0.100E 00
0.100E 02
0.100E-02
LOG CYCLES
AMP RATO N2P3113 100

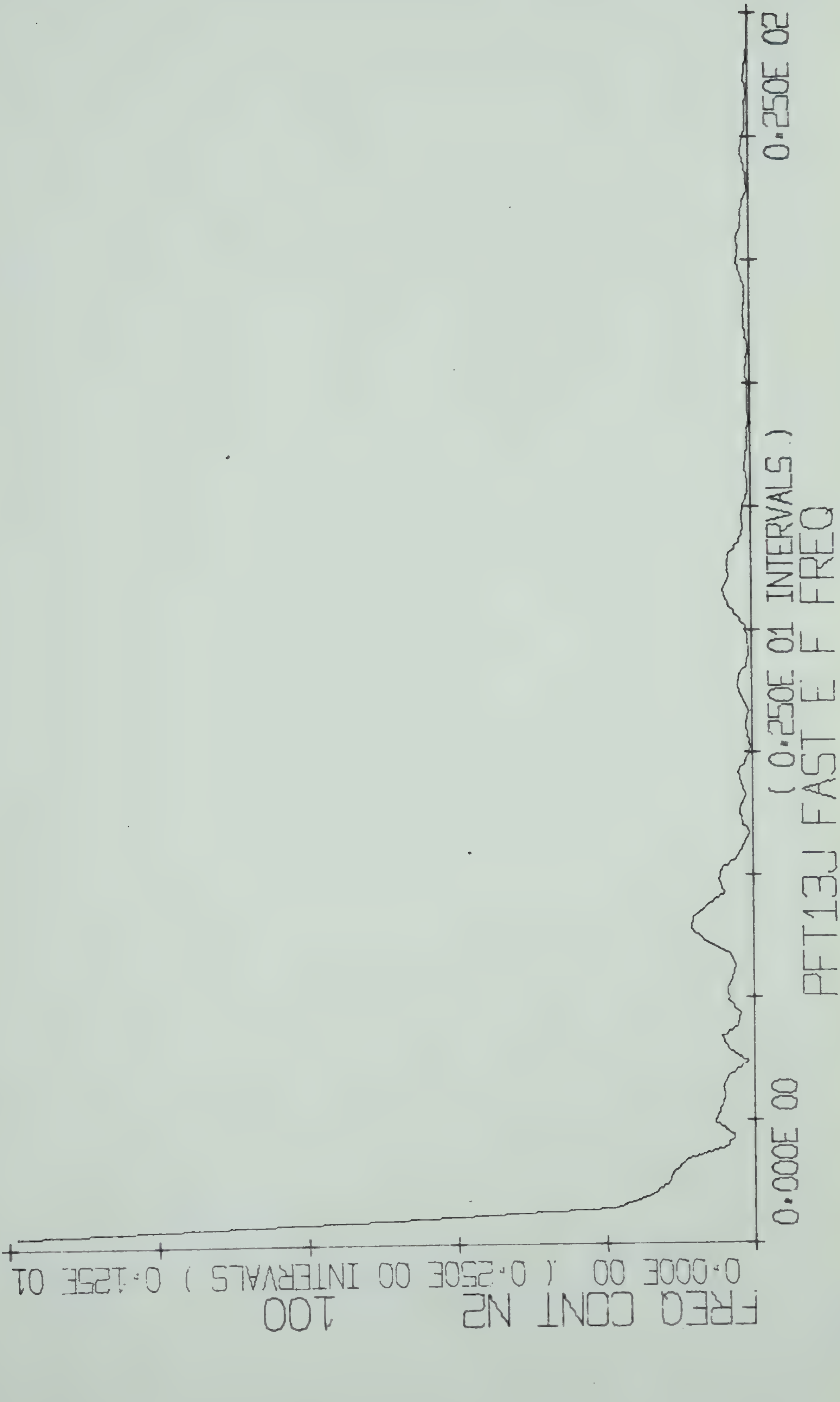

0.100E 02
0.100E-01
LOG CYCLES
PFN13W FILN E R FREQ
0.100E 01
0.100E-01 LOG CYCLES
AMP RATIO RP3113 40

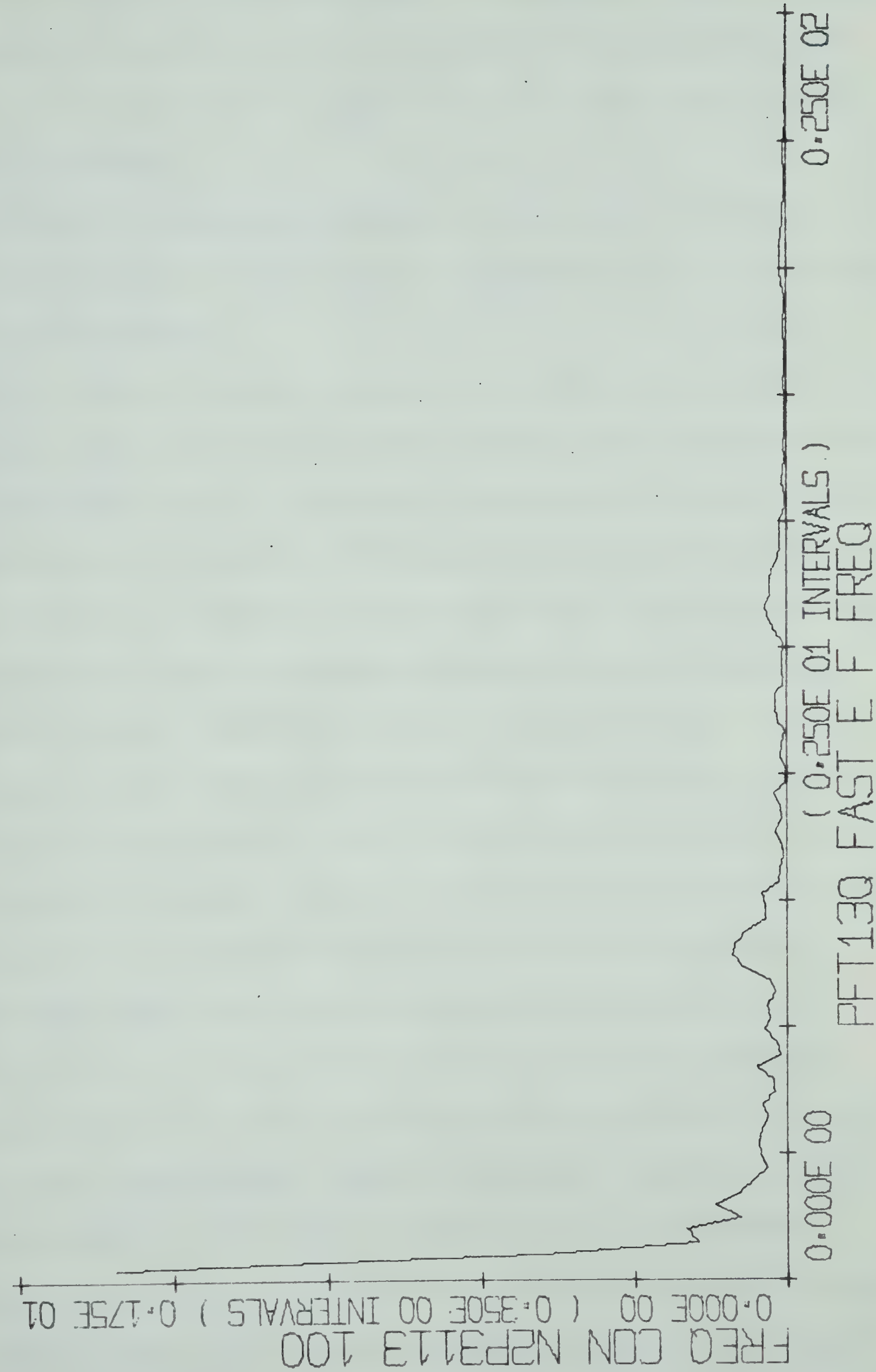

0.000E 00
(0.250E 01 INTERVALS)
0.250E 02
PFT13G FAST E F FREQ
0.000E 00 (0.200E 01 INTERVALS) 0.100E 02
FREQ CONT R
100


PFT13X EXPR E F FREQ
0.000E 00 (0.250E 01 INTERVALS) 0.250E 02
FREQ CONT RP3113 100
0.000E 00 (0.200E 01 INTERVALS) 0.100E 02


0.000E 00
0.250E 02
(0.250E 01 INTERVALS)
PFT13P FAST E F FREQ
0.000E 00 (0.700E 00 INTERVALS) 0.350E 01
FREQ CONT M1P311 100


0.250E 02
(0.250E 01 INTERVALS)
PFT13Z FAST E F FREQ
0.000E 00
0.000E 00 (0.700E 00 INTERVALS) 0.350E 01
FREQ CON M1P3113 100


0.250E 02
(0.250E 01 INTERVALS)
PFT13J FAST E F FREQ
0.000E 00
FREQ CONT N2
100
0.000E 00 (0.250E 00 INTERVALS) 0.125E 01


0.250E 02
(0.250E 01 INTERVALS)
PFT130 FAST E F FREQ
0.000E 00
0.000E 00 (0.350E 00 INTERVALS) 0.175E 01
FREQ CON N2P3113 100

In this work, the error of closure on the pulse curves seemed to be the greatest obstacle to obtaining useful and valid experimental results. In further work every effort should be made to minimize or eliminate this error.

The spectral smoothing, as applied to the theoretical data, showed sufficient promise that further investigation into its usefulness would be well warranted.

In the author's opinion, the use of an on-line data acquisition and control system cannot be considered advantageous for pulse testing, unless exhaustive tests are to be carried on particular process unit as was the intention in this work. In some ways, an on-line digital system can be a hindrance. Firstly, the data acquisition system tends to mask a good deal of the intuitive information which a person testing the process can normally obtain from studying analog recordings. In most of the pulse testing work reported in the literature to date, analog recordings have been used, so there is little information other experiences with digital systems. However, it can be postulated that a lack of analog recordings would be a particular hindrance to intuitive judgment when complicated interacting processes were pulse tested. Secondly, in view of the error of closure problem cited earlier, it would appear desirable in most cases, to study the data at the tail of the pulse and manually select the last value used in the Fourier transform so that the error of closure is exactly zero. With the data acquisition system, it would be possible to develop programs to minimize the error of closure, but it is the author's experience that this would be more work than it is worth, except as mentioned earlier, in the case

of exhaustive pulse tests on a single process unit. Thirdly, in this work it was found that the minimum one second scan cycle of the DDC was not adequate to maintain a steady flow rate after the pulse was completed. This was evidenced by the sharp discontinuities one second apart on the input pulses shown in Appendix II. It is only reasonable to assume that these discontinuities were disturbances to the heat exchanger and, therefore, tended to invalidate the experimental pulse test results.

On the basis of experience gained in this work, the author agrees with Hougen (39) who feels high sensitivity multi-channel analog strip chart recorders and high quality sensors represent the best equipment to use in pulse testing of most industrial processes. Usually a pulse testing program will be a short term study designed to gain specific information about a particular process, so control strategies can be developed before purchasing a data acquisition and control system, such as the one used in this work. A data acquisition and control system is designed to perform routine tasks, something which pulse testing generally does not involve because from one properly executed pulse test, the frequency response characteristics of a process can be obtained. Although it often takes more than one pulse test to get a properly executed pulse test, the task is far from routine. The routine part of pulse testing is the evaluation of the Fourier transforms, something which can be readily done on any general purpose digital computer.

6.5 Future Work

From the knowledge gained in this work, it appears that further study should be considered in the following areas:

a. Spectral smoothing and its usefulness, if any, to pulse testing should be evaluated. In addition, other types of filters should be applied to the calculated frequency data (Section 6.2.3).

b. The problem of an error of closure on the pulse data and the usefulness of the error of closure compensation suggested by Hougen (1964) should be evaluated (Section 6.2.4).

c. The effect of noisy time domain data and the effect of time domain smoothing should be studied.

d. The effect of the time domain sample interval on the position of the amplitude ratio curve should be defined (Sections 6.2.2 and 6.2.5).

e. Further study of pulse shapes is warranted. There is a strong need for some general guidelines which would help the inexperienced person to achieve the ability to conduct successful pulse tests quickly. Also, particular combinations of pulse shapes (single and double-sided) should be studied. The feasibility of using combinations of shapes to obtain frequency information over a range of frequencies which is broader than any range which may be studied using a single pulse should be investigated.

f. Process non-linearities should be studied to determine their effect on frequency response information obtained from pulse tests. The digital non-linear and linearized models for a double pipe heat exchanger, which were derived by Privott (62), could be used as a starting point for this work.

g. An attempt should be made to use pulse testing to reveal the resonance characteristics common to distributed parameter systems

such as the double pipe heat exchanger. The work on resonance of heat exchangers reported in Chapter II could be used as a starting point for this work.

h. Further study of the three Fourier transform techniques used in this work would be valuable. A criterion for the breakdown of the trapezoidal technique at higher frequencies should be established. The programs used for the Filon's and trapezoidal techniques should be modified to allow the use of a variable integration interval on the tail of the pulse and to allow frequency points to be calculated at equal logarithmic increments as well as equal linear increments (Section 6.2.2).

i. Facility should be provided on the DACS to control at a sample rate of less than once per second, say ten times per second. At the once per second rate used in this work, it was found that the flow control was unsatisfactory (Section 6.4) and that the ramp shaped pulses exhibited a staircase effect, (Appendix II). Without the faster sample rate it will be difficult to eliminate the error of closure on pulse data and to generate pulses of different shapes.

j. To aid in improving flow control pressure regulators or pulsation, dampers should be installed in the water lines to the test heat exchanger.

k. A linear frequency to voltage converter would simplify the measurement of the signal from the flow turbines (Section 5.2.2).

BIBLIOGRAPHY

1. Banham, J.W., Jr., "Obtain Process Dynamics by Pulse Testing" Cont. Eng., 12, 83-88, (April 1965)

2. Banham, J.W., Jr., "Development of Experimental Techniques for Frequency Response Analysis by Pulse Test Method", Report of Naval Boiler and Turbine Lab., Philadelphia, Pa., (1964)

3. Banham, J.W., Jr., "Experimental Determination of Open-Loop Frequency Response Characteristics of DLG-9 Class Steam Generator System", Report of Naval Boiler and Turbine Lab., Philadelphia, Pa., (1964)

4. Blackman, R.B., "Linear Smoothing and Prediction in Theory and Practice", 132-136, Addison-Wesley, Reading, Mass., (1965)

5. Blackman, R.B., and Tukey, J.W., "The Measurement of Power Spectra", Dover Publications, N.Y., (1958)

6. Bollay, W., "Aerodynamic Stability and Automatic Control", Journal of the Aeronautical Sciences, 569-617, (Sept. 1951)

7. Bower, J.L., "Introduction to the Design of Servo-mechanisms", John Wiley and Sons, New York, (1958)

8. Bracewell, R., "The Fourier Transform and Its Applications", McGraw-Hill, N.Y., (1965)

9. Brigham, E.O., and Morrow, R.E., "The Fast Fourier Transform", IEEE Spectrum, 63-70, (Dec. 1967)

10. Buckley, P.S., "Techniques of Process Control", John Wiley and Sons, N.Y., 50, (1964)

11. Chan, M., and Leonard, E.F., "Approximate, Simple Transfer Functions for Temperature Forced, Parallel-Flow Heat Exchangers", AIChE Symposium on Systems and Process Control, 56th National Meeting, San Francisco, (May 1965)

12. Clements, W.C., Jr., "Pulse Testing for Dynamic Analysis - Part I: An Investigation of Computational Methods and Difficulties in Pulse Testing; part II: Application of Pulse Testing to Flow and Extraction Dynamics", Ph.D. Thesis, Vanderbilt University, Nashville, Tennessee (1963)

13. Clements, W.C., and Schnelle, K.B., Jr., "Pulse Testing for Dynamic Analysis", I and EC Process Design and Development, 2, No. 2, 94, (1963)

14. Cochran, W.T., and Others, "What is the Fast Fourier Transform?", TEEE Trans. on Audio and Electroacoustics, AV-15, No. 2, 43-117, (June 1967)

15. Cohen, W.C., and Johnson, E.F., "Distributed Parameter Process Synamics", Chem. Eng. Prog., No. 36, 57, (1961)

16. Cohen, W.C., and Johnson, E.F., "Dynamic Characteristics of Double-Pipe Heat Exchangers", Ind. Eng. Chem., 48, 1031, (June 1936)

17. Committee SP26, "Dynamic Response Testing of Process Control Instrumentation", ISA Standard - S26, ISA, Pittsburgh, Pa., (Oct. 1968)

18. Cooley, J.W., "Applications of the Fast Fourier Transform Method", Proc. IBM Scientific Computing Symposium on Digital Simulation of Continuous Systems, 83-96, IBM, White Plains, N.Y., (1967)

19. Cooley, J.W., and Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, 19, No. 90, 297-301, (1965)

20. Coppedge, T.N., Jr., Wolf, R.L.,Rept. T-68, Instrumentation Laboratory, Mass. Inst. Technol., (May 1954) (Unclassified, available from ASTIA, Dayton, Ohio)

21. Debolt, R.R., "Dynamic Characteristics of a Steam Water Heat Exchanger", M.Sc. Thesis, University of California, Berkeley, California, (1954)

22. Draper, C.S., McKay, W., and Lees, S., "Instrument Engineering", II, Chapt. 25, 602, 654-656, 660-661, McGraw-Hill (1953)

23. Dreifke, G.E., and Hougen, J.O., "Process Diagnosis and Model Formulation by Pulse Methods", 81-113, Instrumentation in the Chemical and Petroleum Industries, 2, Publication of ISA edited by C. W. Sanders, Plenum Press, N.Y., (1966)

24. Dreifke, G.E., Hougen, J.O., and Mesmer, G., "Effects of Truncation on Time to Frequency Domain Conversion", ISA Trans., 1, No. 4, 353-368, (Oct. 1962)

25. Dreifke, G.E., "Effects of Input Pulse Shape and Width on Accuracy of Dynamic System Analysis from Experimental Pulse Data", Ph.D. Thesis, Washington University, Sever Institute of Technology, St. Louis, Missouri (1961)

26. Edwards, C.L., "Dynamics of Flow-Forced Concentric Tube Heat Exchanger", Ph.D. Thesis, University of Wisconsin (1960)

27. Filon, L.N.G., "On a Quadrature Formula for Trigonometric Integrals", Proc. Roy, Soc., Edinburgh, 49, 38, (1928)

28. Fisher, D.G., "Frequency Response of Flow Forced Heat Exchangers", Ph.D. Thesis, University of Michigan, (1965)

29. Fogle, J.B., "Sieve Plate Distillation Dynamics", Ph.D. Thesis, Virginia Polytechnic Institute, Blacksburg, Virginia (1966)

30. Gaskell, R.E., "Engineering Mathematics", Holt, Rinehart and Winston, N.Y., (1958)

31. Gentlemen, W.M., and Sande, G., "Fast Fourier Transforms - For Fun and Profit", 1966 Fall Joint Conf. AFIPS Proc., 29, Washington, D.C., Spartan, 563-578, (1966)

32. Go, R.E., "Pulse Testing of a Multicomponent Distillation Column", M.Sc. Thesis, Virginia Polytechnic Institute, Blacksburg, Virginia, (1965)

33. Grantom, R.L., Hougen, J.O., and Driefke, G.E., "Stream Analyzer Dynamics","Part I: Testing Techniques and Apparatus", Cont. Eng., 7, 113-115, (May 1960), "Part II: Refractometer Response to Temperature and Composition Change", Cont. Eng., 7, 104-108, (July 1960)

34. Harriott, P., "Process Control", McGraw-Hill, N.Y., (1964)

35. Head, F.E., Hougen, J.O., Walsh, R.A., "Determining the Properties of Continuous Flow Systems by Pulse Excitations", Proc. of the International Federation of Automatic Control Cong., Moscow, Butterworths Scientific Publications, London, 495, (1960)

36. Hearn, J.S., "The Dynamic Response of a Heat Exchanger to Shell-Side Flow Rate Disturbances", M.Sc. Thesis, Washington University, St. Louis, (1959)

37. Herron, E.H., Jr., "Distributed Parameter System Dynamics, The Double-Pipe Heat Exchanger", Ph.D. Thesis, Tulane University, New Orleans, Louisiana

38. Hemple, A., "On the Dynamics of Steam Liquid Heat Exchangers", Trans. ASME, 82, 244-251, (1961)

39. Hougen, J.O., "The Experimental Approach to Process Control Problems", Presented at the Fourth Annual Conference on the Use of Digital Computers in Process Control, Louisiana State University, Baton Rouge, Louisiana, (1968)

40. Hougen, J.O., "Experience and Experiments with Process Dynamics", Chem. Eng. Progr. Monograph Series, No. 4, 60, (1964)

41. Hougen, J.O., "Process Dynamics by Pulse Testing", Chem. Eng., 68, 209-212, (June 12, 1961)

42. Hougen, J.O., and Walsh, R.M., "Pulse Testing Method", Chem. Eng. Progr., 57, No. 3, 69, (1961)

43. Hougen, J.O., "Methods of Procuring Frequency Response Data in the Chemical Industry", Proceedings of the Seminar in Systems, edited by G. D. S. MacLellan and Samuel Mercer, Jr., Michigan State University, 127-142, (1958)

44. Hsu, J.P., and Gilbert, N., "Transfer Functions of Heat Exchangers", AIChE Journal, No. 5, 8, 593-598, (1962)

45. Hutchinson, A.W., and Shelton, R.J., "Measurement of Dynamic Characteristics of a Full-Scale Plant Using Random Perturbing Signals: An Application to a Refinery Distillation Column", Trans. Inst. Chem. Engrs., 45, T334-T342, (1967)

46. Isom, K.G., "The Effects of Flow Variations on the Dynamics of a Liquid - Liquid Heat Exchanger", Ph.D. Thesis, Northwestern University, (1963)

47. Janis, R.F., "Feedforward Control of Binary Distillation", Ph.D. Thesis, Virginia Polytechnic Institute, Blacksburg, Virginia (1967)

48. Jenkins, G.M., and Watts, D.G., "Spectral Analysis and Its Applications", Holden-Day, San Francisco, (1968)

49. Koppel, L.B., "Dynamics of a Flow-Forced Heat Exchanger", Ind. Eng. Chem. Fund., No. 2, 1, 131-134, (1962)

50. Lees, S., and Dougherty, R.C., "Numerical Fourier Transform Calculations for Pulse Testing Procedures", ASME, Journal of Basic Engineering, 445-449, (1967)

51. Lees, S., "Interpreting Dynamic Measurements of Physical Systems", Trans. ASME, 833-857, (May 1958)

52. Lees, S., and Hougen, J.O., "Pulse Testing a Model Heat Exchanger Process", Ind. Eng. Chem., 48, 1064-1068, (1956)

53. Marino, P.A., Perna, A.J. and Stutzman, L.F., "Sinusoidal and Pulse Response of a Plate Distillation Column by Reflux Upset", AIChE Journal, No. , 36, 866-869, (1968)

54. Marino, P.A., and Stutzman, L.F., "Distillation Column Dynamics With the Use of the Pulse Technique", AIChE Journal, No. 3, 12, 603-604, (1966)

55. Ming, L.L., "Numerical Techniques of Fourier Transforms with Applications", Proc. of Second Annual Allerton Conf. on Circuit and System Theory, 114-134, (1964)

56. Monroe, A.J., "Digital Processes for Sampled Data Systems", John Wiley and Sons, N.Y., (1962)

57. Morris, H.J., "The Dynamic Response of a Heat Exchanger to Inlet Temperature Disturbances", M.Sc. Thesis, St. Louis University, (1959)

58. Mozley, W.H., "Predicting Dynamics of Concentric Pipe Heat Exchangers", Ind. and Eng. Chem. No. 6, 48, 1035-1041, (June 1956)

59. Nyquist, J.K., Schindler, R.N., and Gilbert, R.E., "Determination of Frequency Response From Step Response", Chem. Eng. Progr., Symposium Series, 46, 59, 98-105, (1963)

60. Papoulis, A., "The Fourier Integral and Its Applications", McGraw-Hill, N.Y., (1962)

61. Paynter, H.M., and Takohoshi, Y., "A New Method of Evaluating Dynamic Response of Counter-Flow and Parallel Flow Heat Exchangers", Trans. ASME, No. 4, 78, 749, (May 1956)

62. Privott, W.J., Jr., "Analysis of Flow Forced, Concentric Tube Heat Exchangers", Ph.D. Thesis, University of North Carolina, North Carolina, (1965)

63. Raven, F.H., "Mathematics of Engineering Systems", McGraw-Hill, N.Y., (1966)

64. Rea, J.B., "Dynamic Analysis of Aeroelastic Aircraft by the Transfer Function - Fourier Method", Journal of the Aeronautical Sciences, 375, (June 1951)

65. Renfroe, C.A., Jr., "Multicomponent Distillation Dynamics", Ph.D. Thesis, Virginia Polytechnic Institute, Blacksburg, Virginia, (1965)

66. Ridings, R.V., "Errors in Experimental Evaluation of System Frequency Response", M.Sc. Thesis, St. Louis University, Missouri, (1961)

67. Rosenbrock, H.H., "The Transient Behaviour of Distillation Columns and Heat Exchangers. A Historical and Critical Review", Trans. Instn. Chem. Engrs., 40, (1962)

68. Sanborn, I.B., and Chao, H., "The Identification of Process Dynamics", Presented at the International Symposium on Pulp and Paper Process Control, Co-sponsored by the ISA and CPPA, (1969)

69. Savitzky, A., and Golay, M.J.E., "Smoothing and Differentiation of Data by Simplified Least Squares Procedures", Analytical Chemistry, No. 8, 36, 1627 - 1639, (July 1964)

70. Schechter, R.S., and Wissler, E.H., "Frequency Response From Step Input Response", Ind. Eng. Chem., 51, 945 (1959)

71. Seamans, R.C., Jr., Blasingame, B.P., and Clementson, G.C., "The Pulse Method for the Determination of Aircraft Dynamic Performance", Journal of the Aeronautical Sciences, 22-38, (January 1950)

72. Shinskey, F.G., "Process-Control Systems", McGraw-Hill, (1967)

73. Smith, G.A., and Triplett, W.C., "Experimental Flight Methods for Evaluating Frequency-Response Characteristics of Aircraft"

74. Stermole, F.J., and Larson, M.A., "The Dynamics of Flow-Forced Distributed Parameter Heat Exchangers", AIChE Journal, No. 5, 10, 688-694, (1964)

75. Stermole, F.J., "Dynamic Response of Flow-Forced Heat Exchangers", Ph.D. Thesis, Iowa State University, (1963)

76. Thal-Larson, H., "Dynamics of Heat Exchangers and Their Models", Journal of Basic Eng., Trans. ASME, 82, 489-504, (June 1960)

77. Vincent, G.C., Hougen, J.O., and Dreifke, G.E., "Fluid Mixing in Shell and Tube Heat Exchangers", ASME Paper, No. 60-WA-261

78. Wollaston, E.G., and Swanson, B.S., "A Study of Pulse, Step, and Random Signal Analysis of Systems", Systems and Process Control, A CEP technical manual published by AIChE, N.Y., (1967)

79. Walters, E.R., and Rea, J.B., "Determination of Frequency Characteristics From Response to Arbitrary Input", Journal of the Aeronautical Sciences, 446-452, (July 1950)

80. Williams, T.J., and Morris, H.J., "A Survey of the Literature on Heat Exchanger Dynamics and Control", Chem. Eng. Progr. Symposium Series, No. 36, 57, 20, (1961)

81. Wylie, C.R., Jr., "Advanced Engineering Mathematics", McGraw-Hill, N.Y., (1960)

NOMENCLATURE

The nomenclature which appears at only one location in the thesis is defined at that point and does not appear in this list.

MATHEMATICAL SYMBOLS

a - general lower limit on integration

$AR()$ - amplitude ratio

b - general upper limit on integration

d - duration of a pulse (sec.)

e - natural logarithm base

f - function of time (general pulse curve)

$F[\ ]$ - indicates Fourier transform operation

$G()$ - transfer function of process

j - -1

s - Laplace transform operator

t - time (sec.)

Δt - sample interval (sec.)

T - duration of a pulse (sec.)

W - frequency (rad./sec.)

X - input pulse variable or subscript implying reference to input variable

y - output variable or subscript implying reference to output variable

$\mathcal{L}()$ - indicates Laplace transform operation

$\phi()$ - phase angle

π - constant (3.1416)

ABBREVIATIONS

ADC - analog to digital converter

COS - current output station

DAC - digital to analog converter

DACS - data acquisition and control system

DAR - data acquisition rapid

DAS - data acquisition slow

DDC - direct digital control

FFT - Fast Fourier transform

FT - results obtained using FFT

FN - results obtained using Filon's quadrature

M - multiplexer

MSSS - mainline subroutine service set

PA - process actuator

POC - process operators console

PT - process transmitter

PVRT - process variable record table

TB - terminal board

TP - results obtained using the trapezoidal quadrature

APPENDIX I

THEORETICAL PULSE GRAPHS

General Comments

a. Table I A describes the graphs presented in this appendix in the order in which they appear.

b. A description of the code used for labelling the axes of the graphs was given in Section 6.1.

c. The data presented in these graphs were obtained using the techniques outlined in Chapter IV.

d. All input pulses are rectangular in shape and all output pulses are derived from the response of a first order system with a time constant of 5.0 seconds.

e. All input and output pulses are on the same scale to facilitate comparisons.

f. All frequency content graphs have the same scale on the abscissa but not necessarily on the ordinate.

g. All frequency content curves were calculated using the Fast Fourier transform.

h. For all runs, the sample interval for the time data was 0.1 seconds except for run 06 where an interval of 0.2 seconds was used.

i. The graphs associated with each run are arranged in the following order:

1. input pulse

2. frequency content of input pulse

3. output pulse

j. The runs are arranged in order from the narrowest to the widest input pulse.

TABLE I A

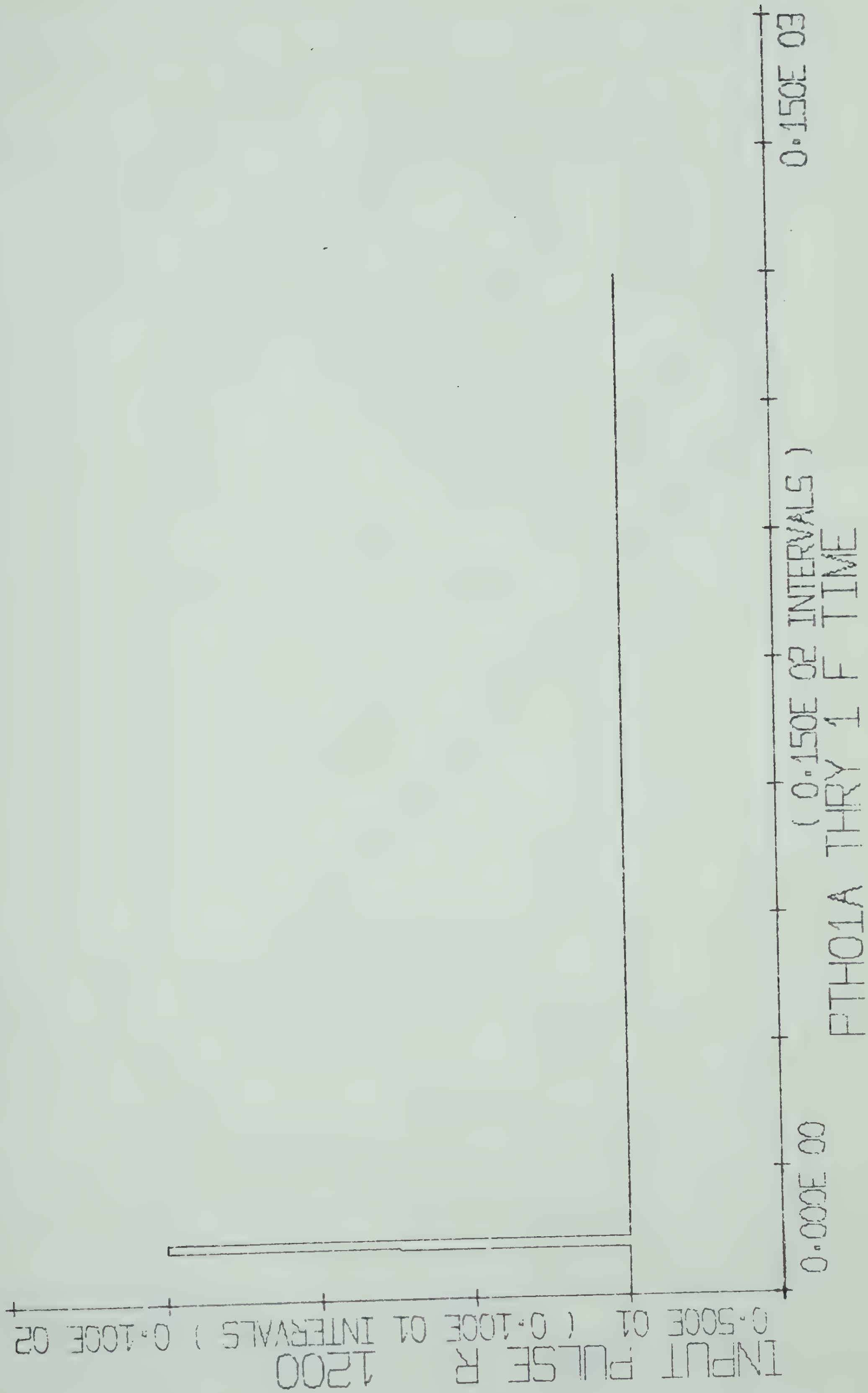
LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR THEORETICAL RUNS

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	COMMENT
PTH01A	275	1.0	Input pulse; rectangular.
PFT01C	276	1.0	Frequency content curve for input pulse PTH01A.
PTH01B	277	1.0	Output pulse; response to input pulse PTH01A.
PTH02A	278	2.5	Input pulse; rectangular.
PFT02C	279	2.5	Frequency content curve for input pulse PTH02A.
PTH02B	280	2.5	Output pulse; response to input pulse PTH02A.
PTH03A	281	5.0	Input pulse; rectangular.
PFT03C	282	5.0	Frequency content curve for input pulse PTH03A.
PTH03B	283	5.0	Output pulse; response to input pulse PTH03A.
PTH04A	284	10.0	Input pulse; rectangular.
PFT04C	285	10.0	Frequency content curve for input pulse PTH04A.
PTH04B	286	10.0	Output pulse; response to input pulse PTH04A.

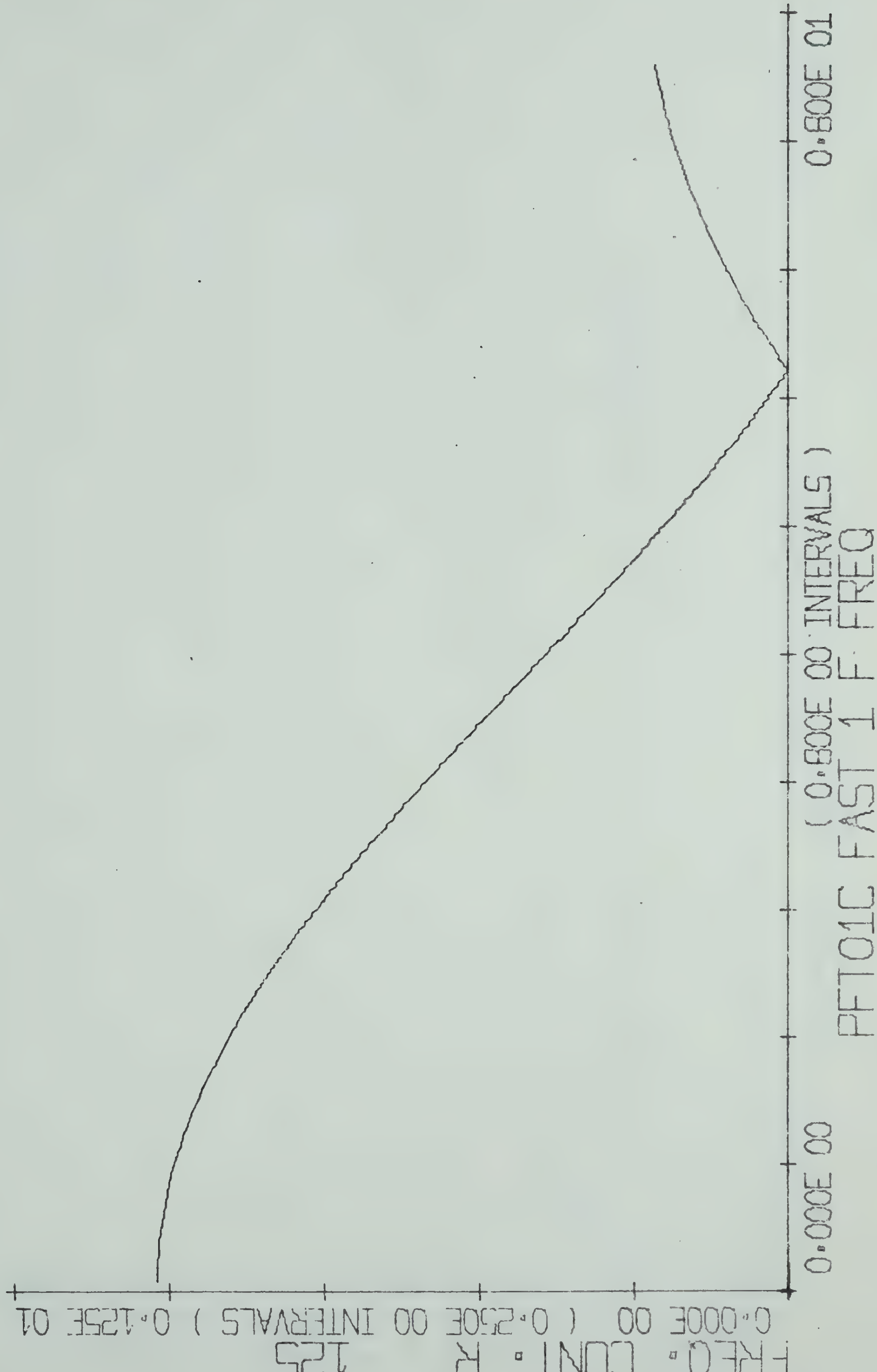
TABLE I A (CONT'D)

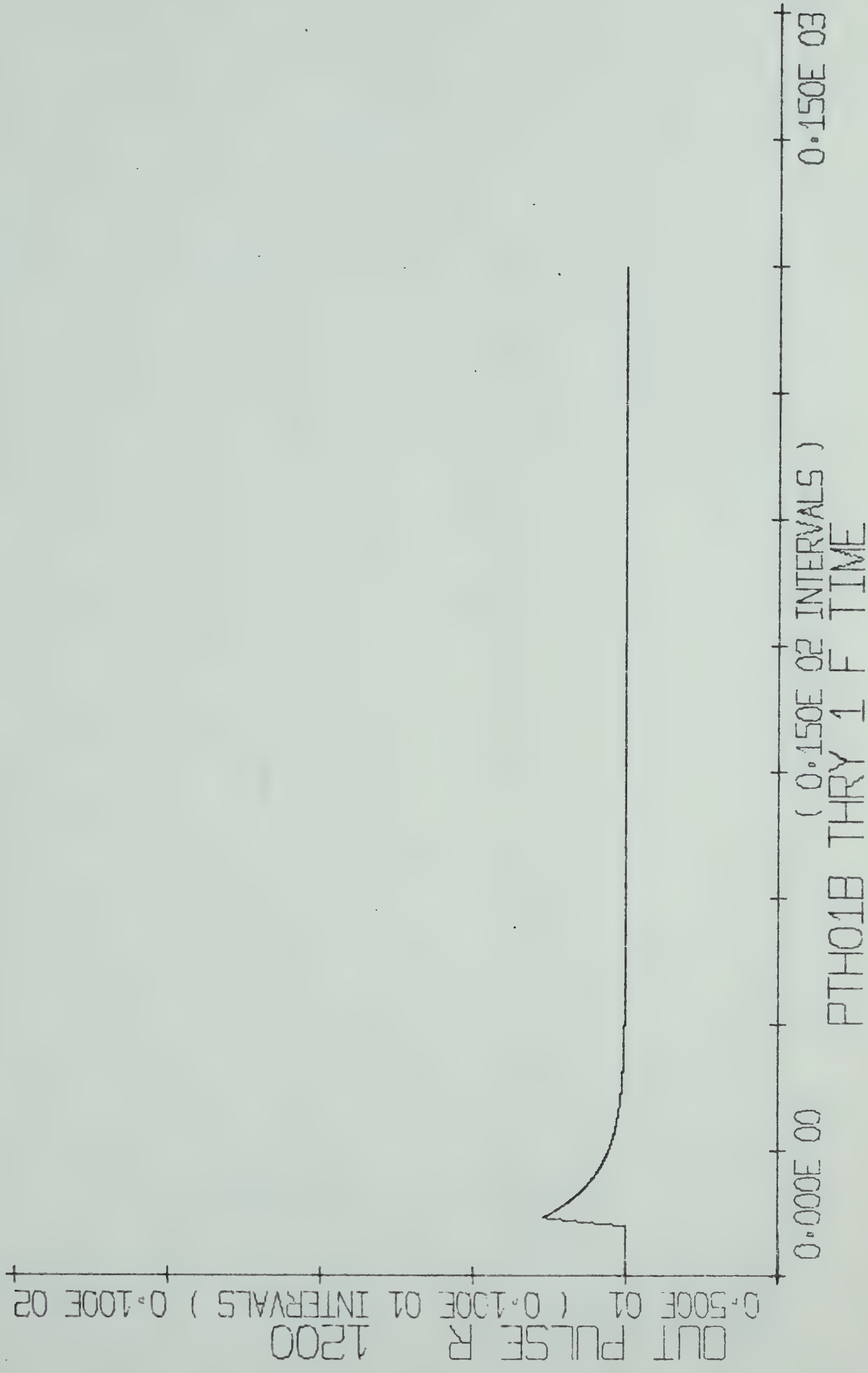
LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR THEORETICAL RUNS

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	COMMENT
PTH05A	287	25.0	Input pulse; rectangular.
PFT05C	288	25.0	Frequency content curve for input pulse PTH05A.
PTH05B	289	25.0	Output pulse; response to input pulse PTH05A.
PTH06A	290	75.0	Input pulse; rectangular.
PFT06C	291	75.0	Frequency content curve for input pulse PTH06A.
PTH06B	292	75.0	Output pulse; response to input pulse PTH06A.



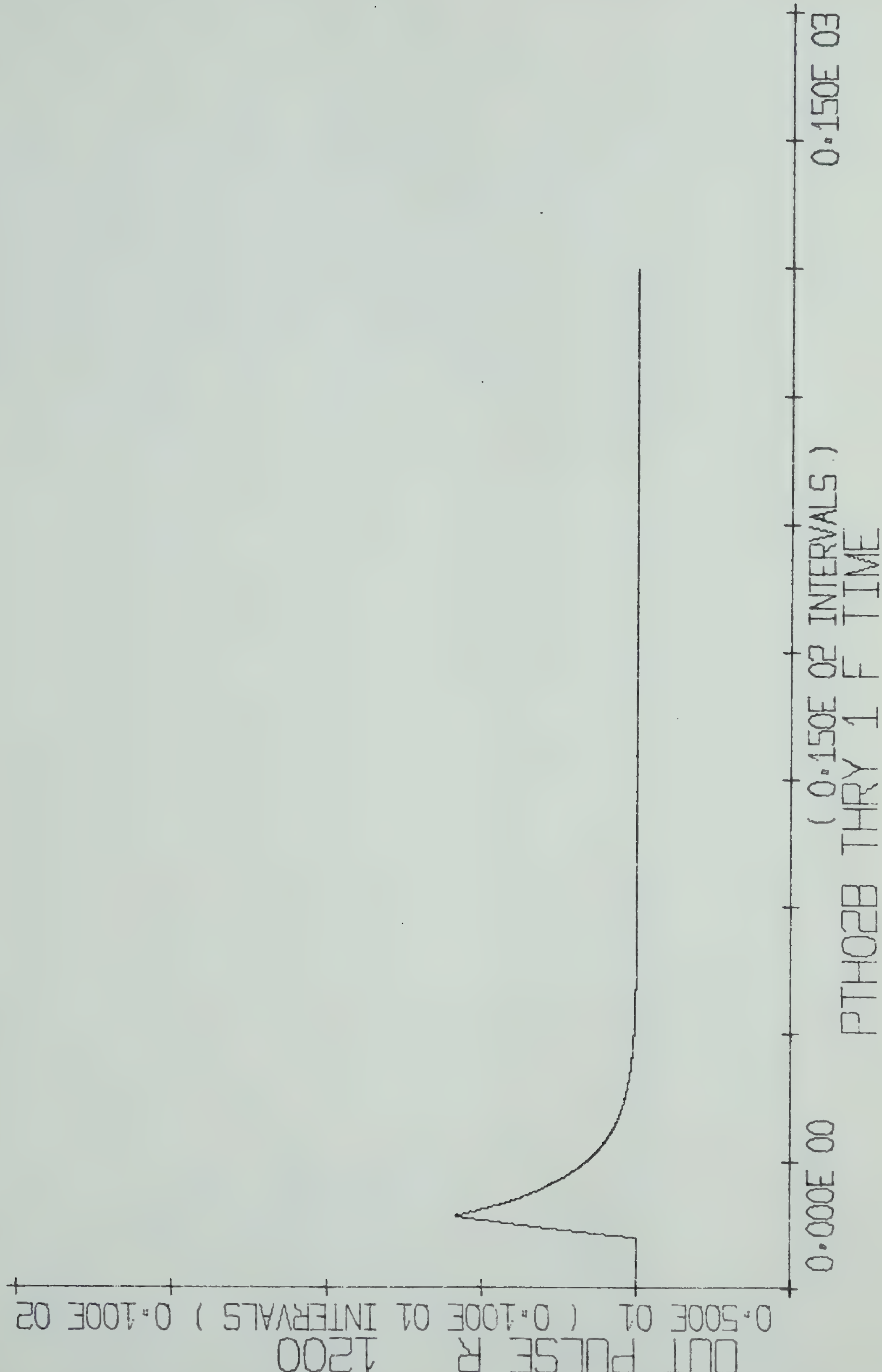
0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
FTH01A THRY 1 F TIME
0.500E 01
(0.100E 01 INTERVALS)
0.100E 02
INPUT PULSE R 1200

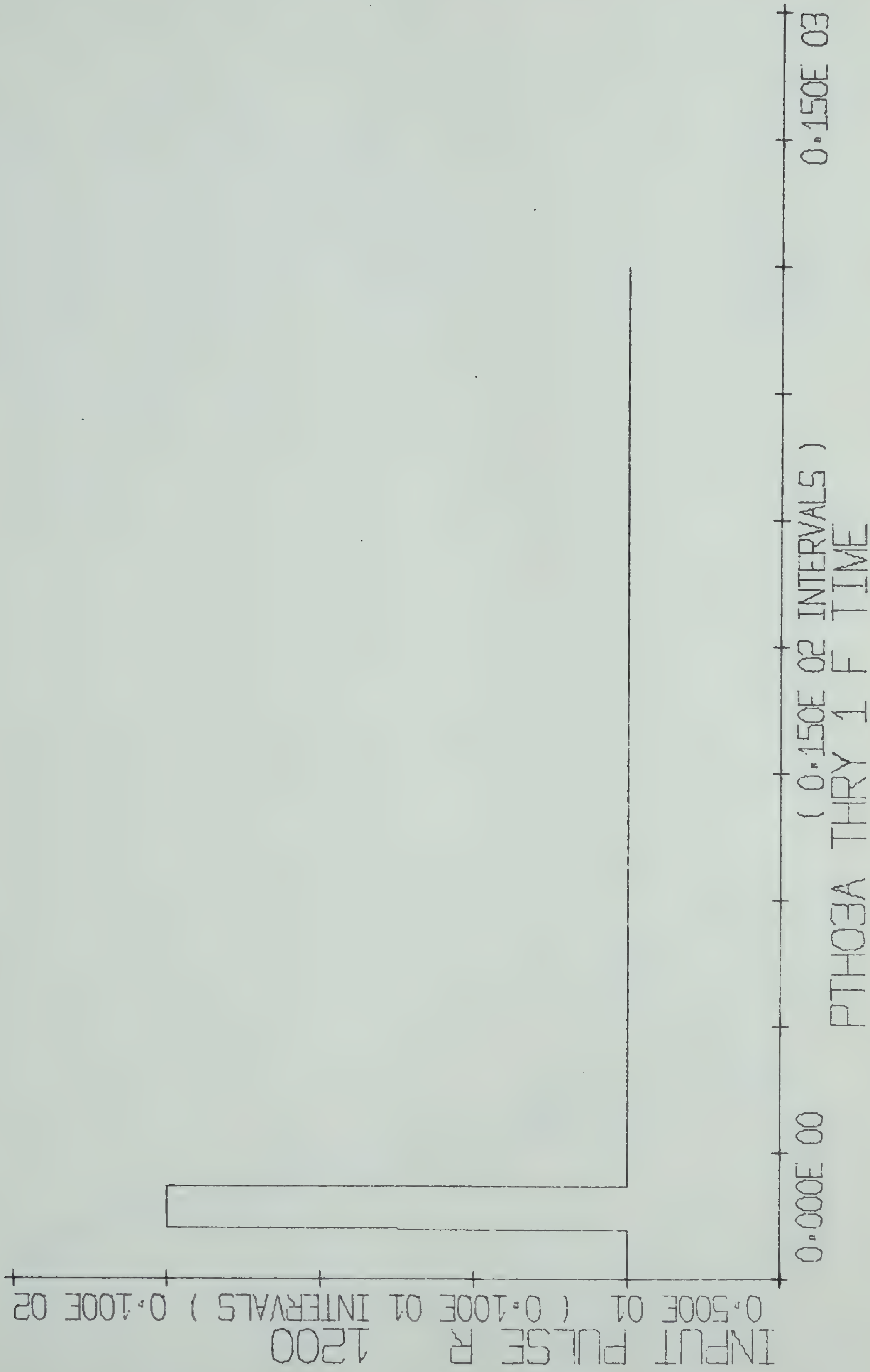

0.800E 01
(0.800E 00 INTERVALS)
PFT01C FAST 1 F FREQ
0.000E 00
0.000E 00 (0.250E 00 INTERVALS) 0.125E 01
FREQ. CONT. R 125


0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
PTHO1B THRY 1 F TIME
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
OUT PULSE R 1200

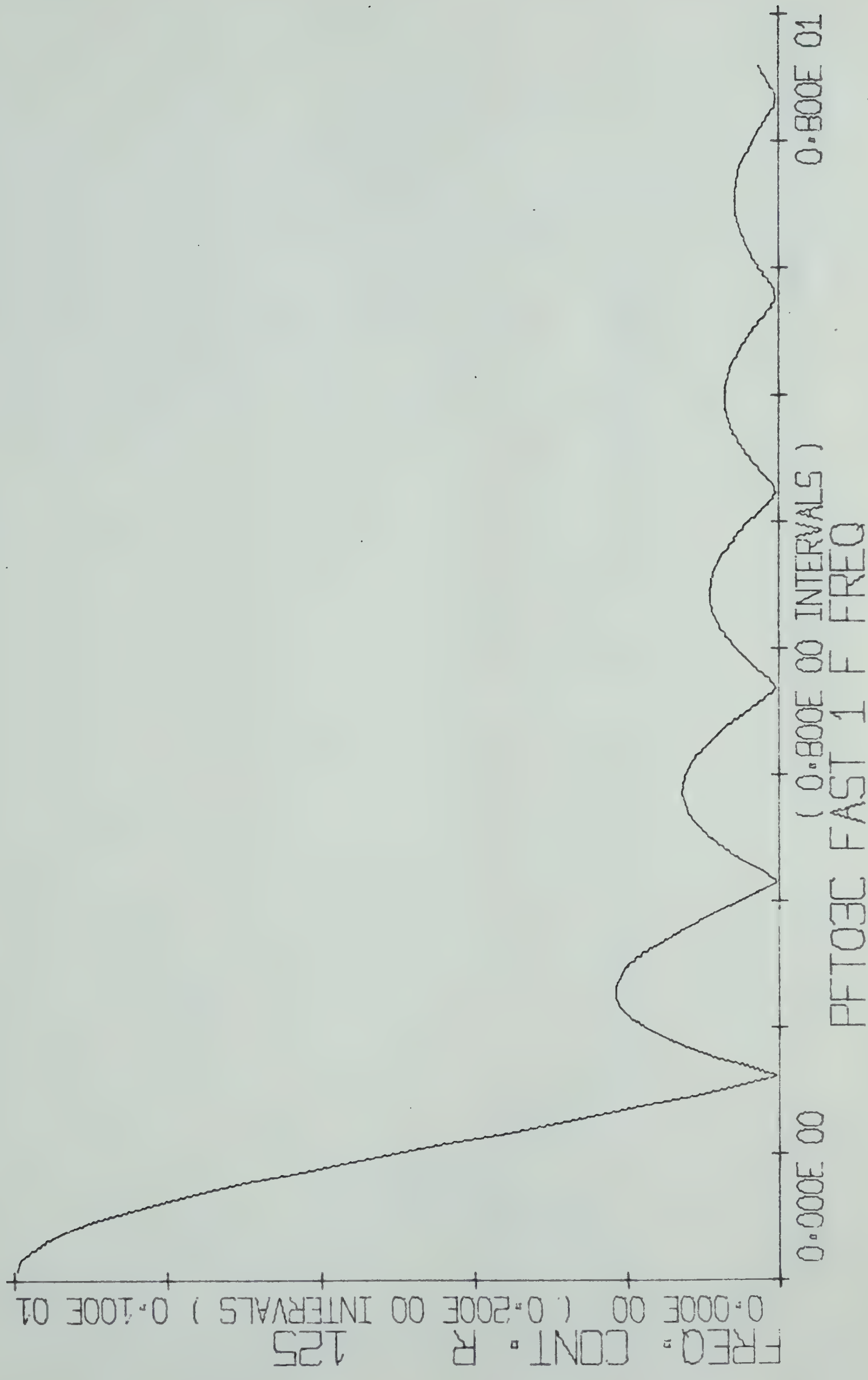
0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
PTHO2A THRY 1 F TIME
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
INPUT PULSE R 1200


0.000E 00
(0.800E 00 INTERVALS)
0.800E 01
PFT02C FAST 1 F FREQ
0.000E 00
(0.200E 00 INTERVALS)
0.100E 01
FREQ. CONT. R 125

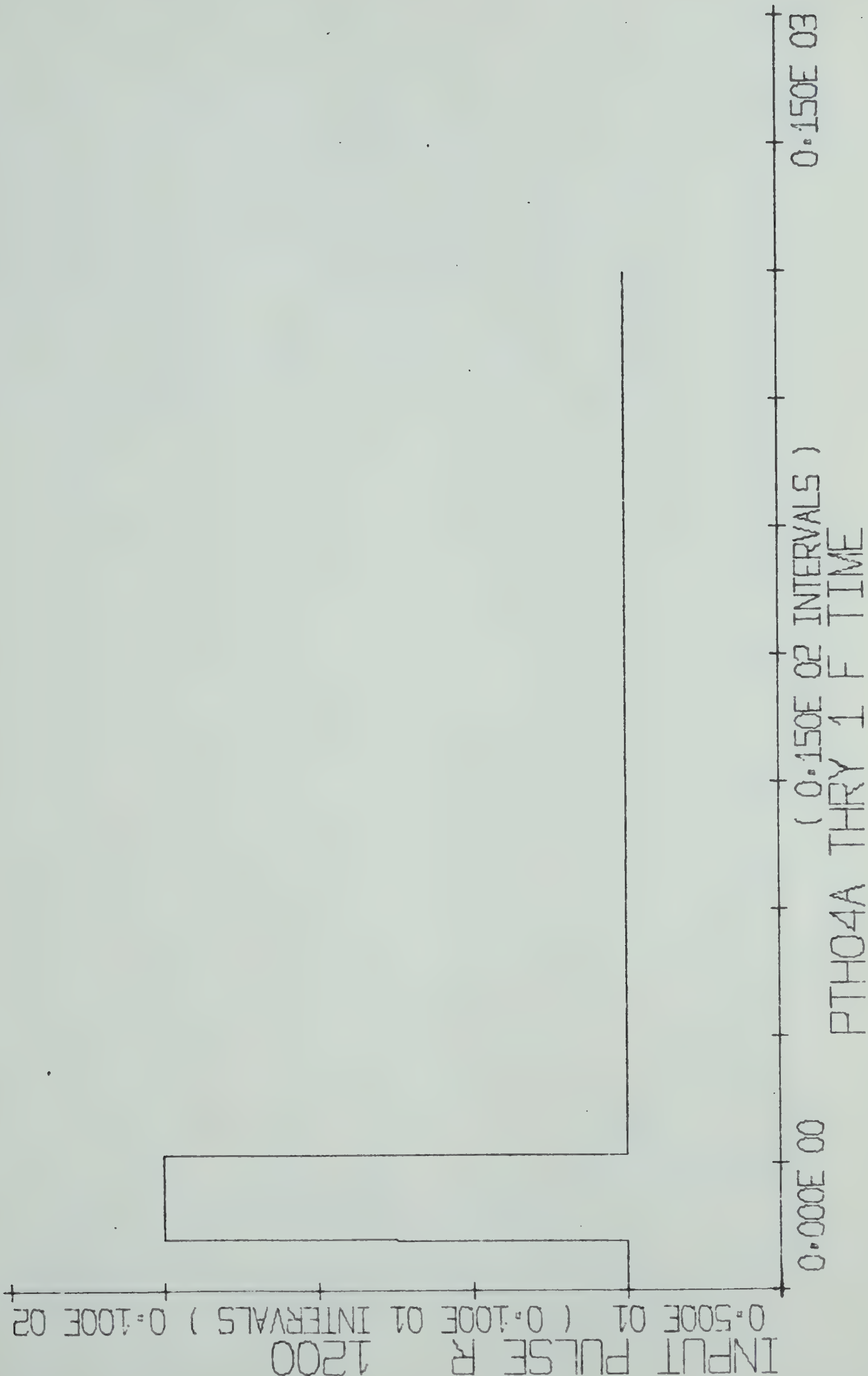

0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
PTHO2B THRY 1 F TIME
0.500E 01
(0.100E 01 INTERVALS)
0.100E 02
OUT PULSE R 1200

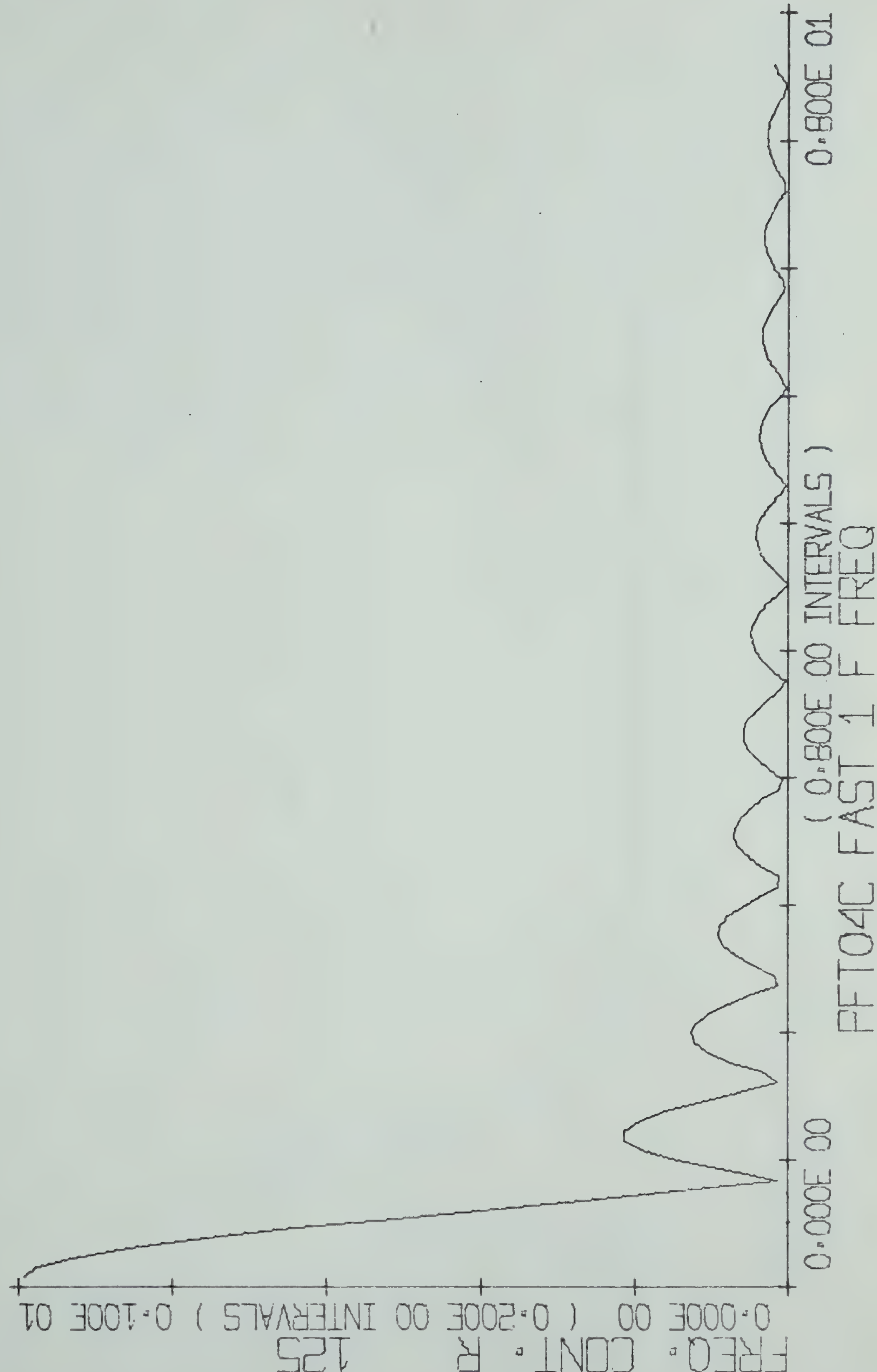


0.150E 03
(0.150E 02 INTERVALS)
PTHO3A THRY 1 F TIME
0.000E 00
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
INPUT PULSE R 1200

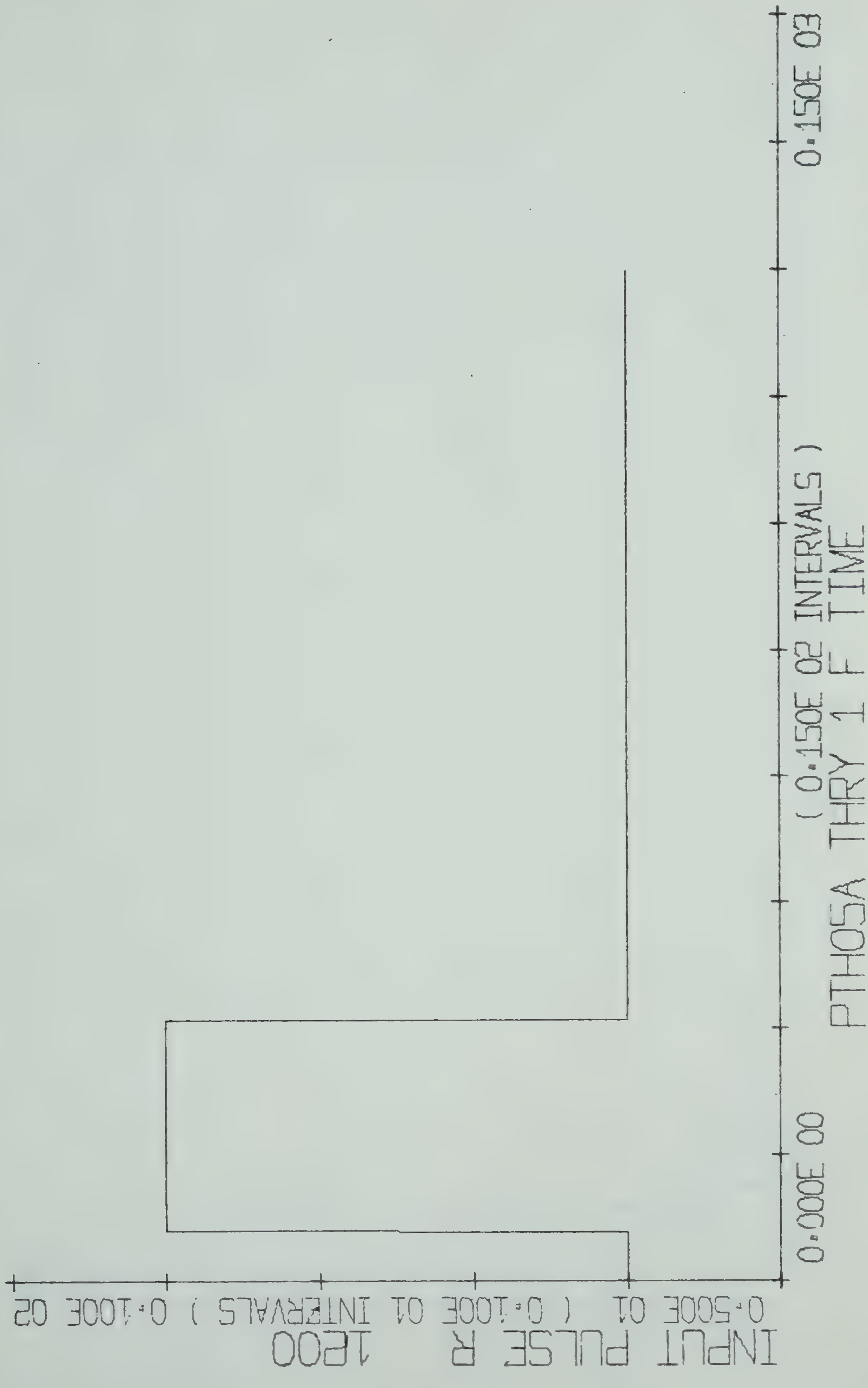

0.000E 00
(0.800E 00 INTERVALS)
0.800E 01
PFT03C FAST 1 F FREQ
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01
FREQ. CONT. R
125

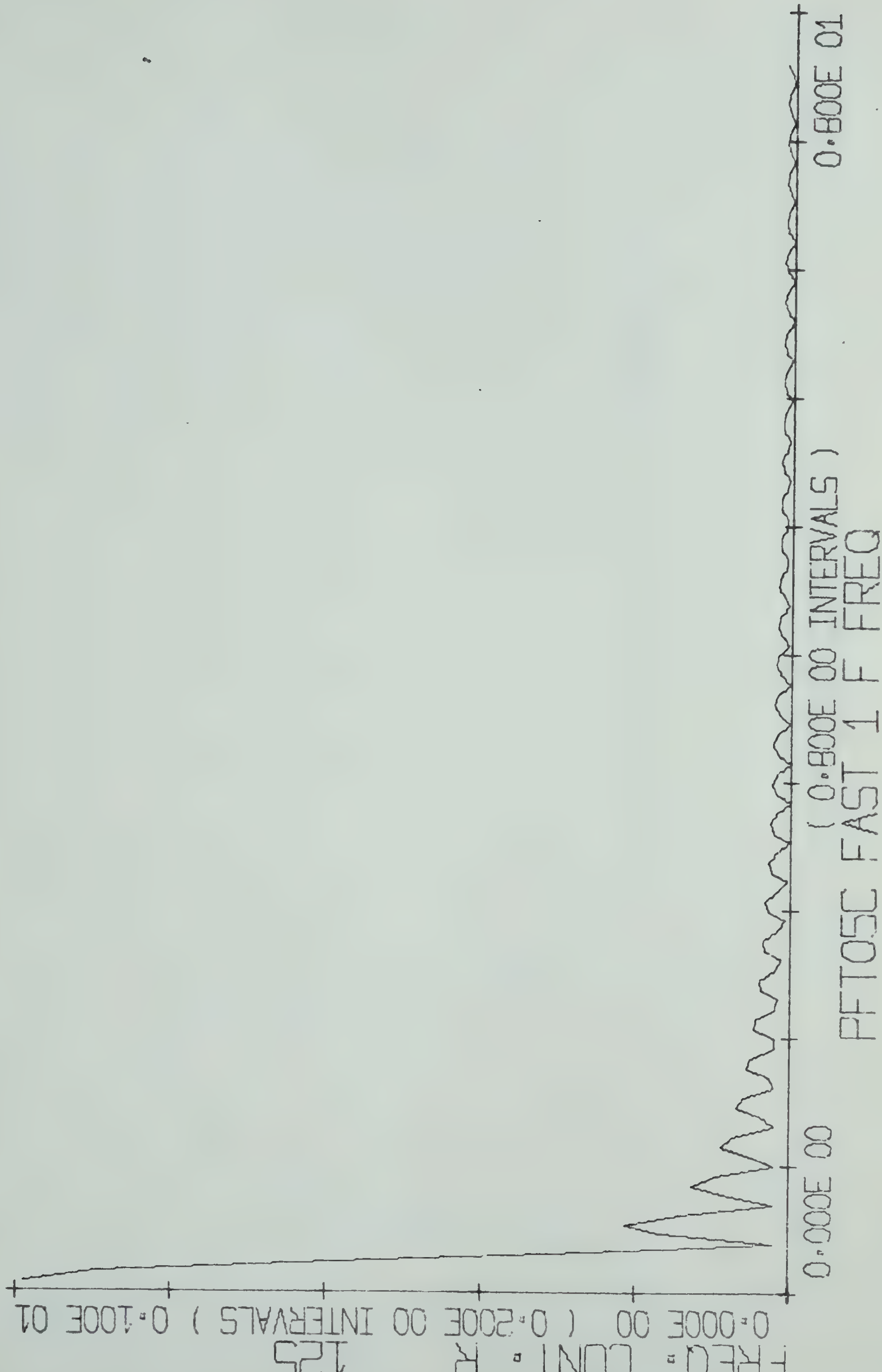

PTHO3B THRY 1 F TIME
(0.150E 02 INTERVALS)
0.000E 00
0.150E 03
OUT PULSE R 1200
0.500E 01
(0.100E 01 INTERVALS)
0.100E 02

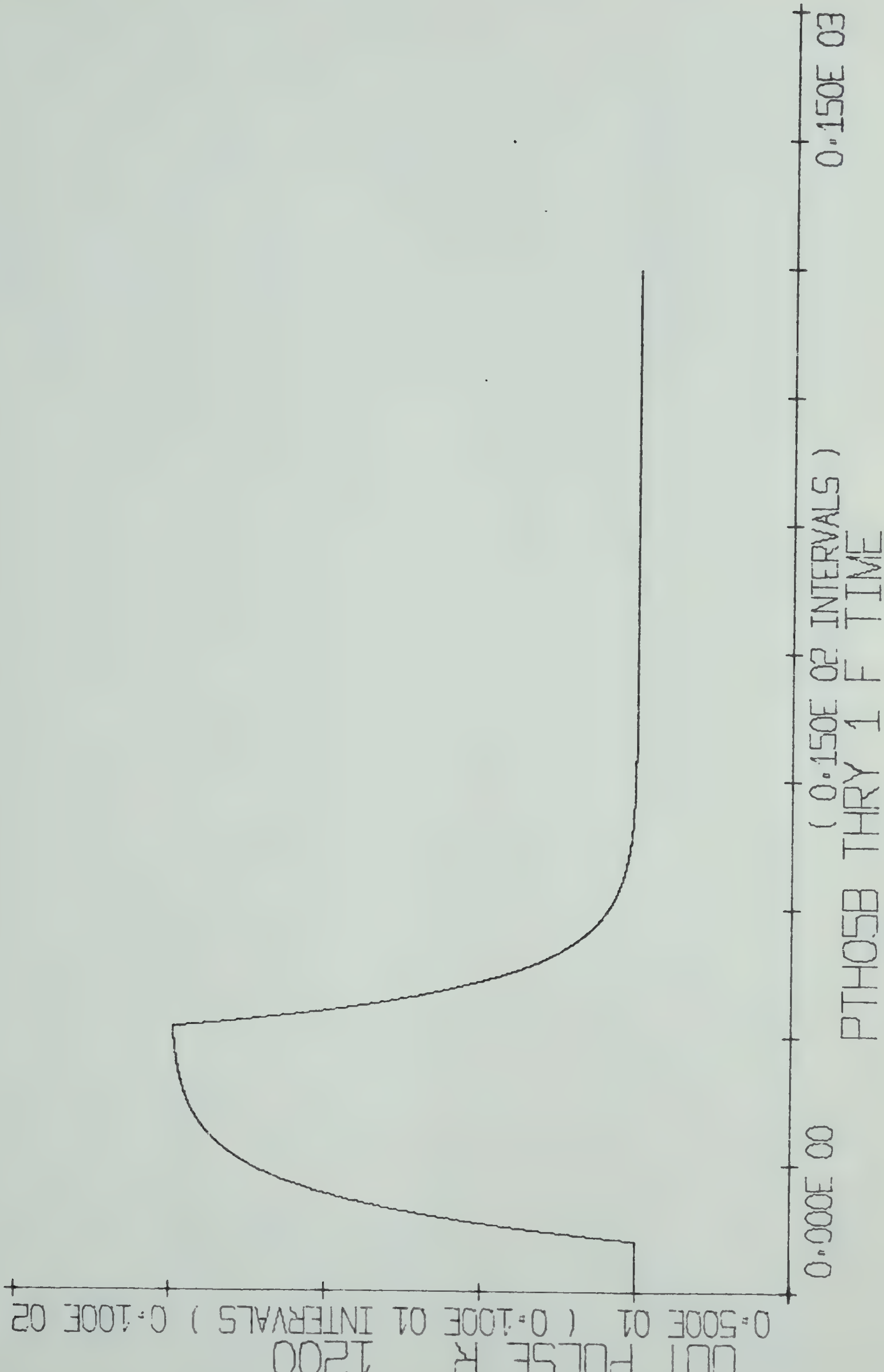

0.150E 03
(0.150E 02 INTERVALS)
PTHO4A THRY 1 F TIME
0.000E 00
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
INPUT PULSE R 1200

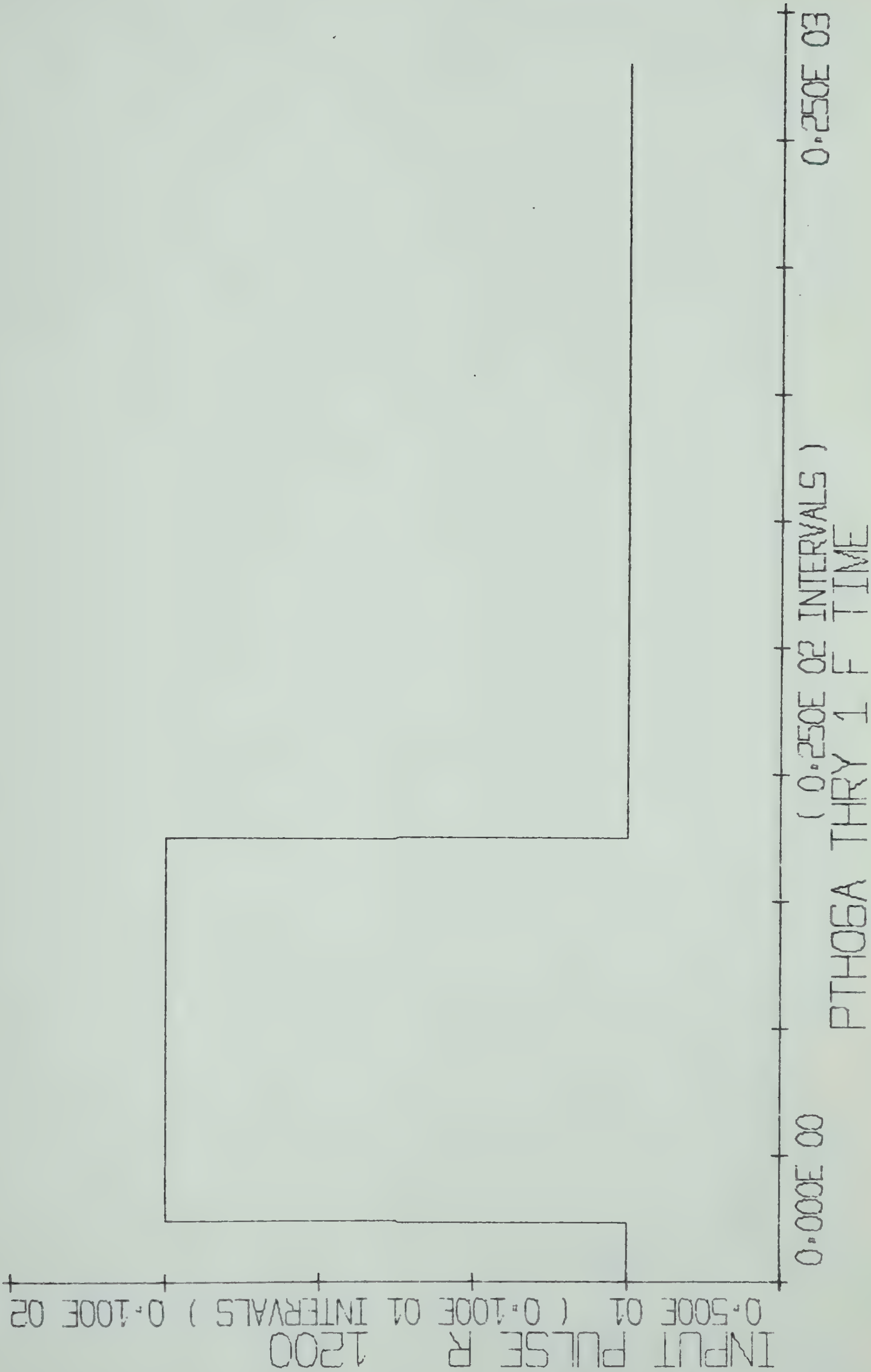

0.000E 00
(0.800E 00 INTERVALS)
0.800E 01
PFTO4C FAST 1 F FREQ
FREQ. CONT. R 125
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01

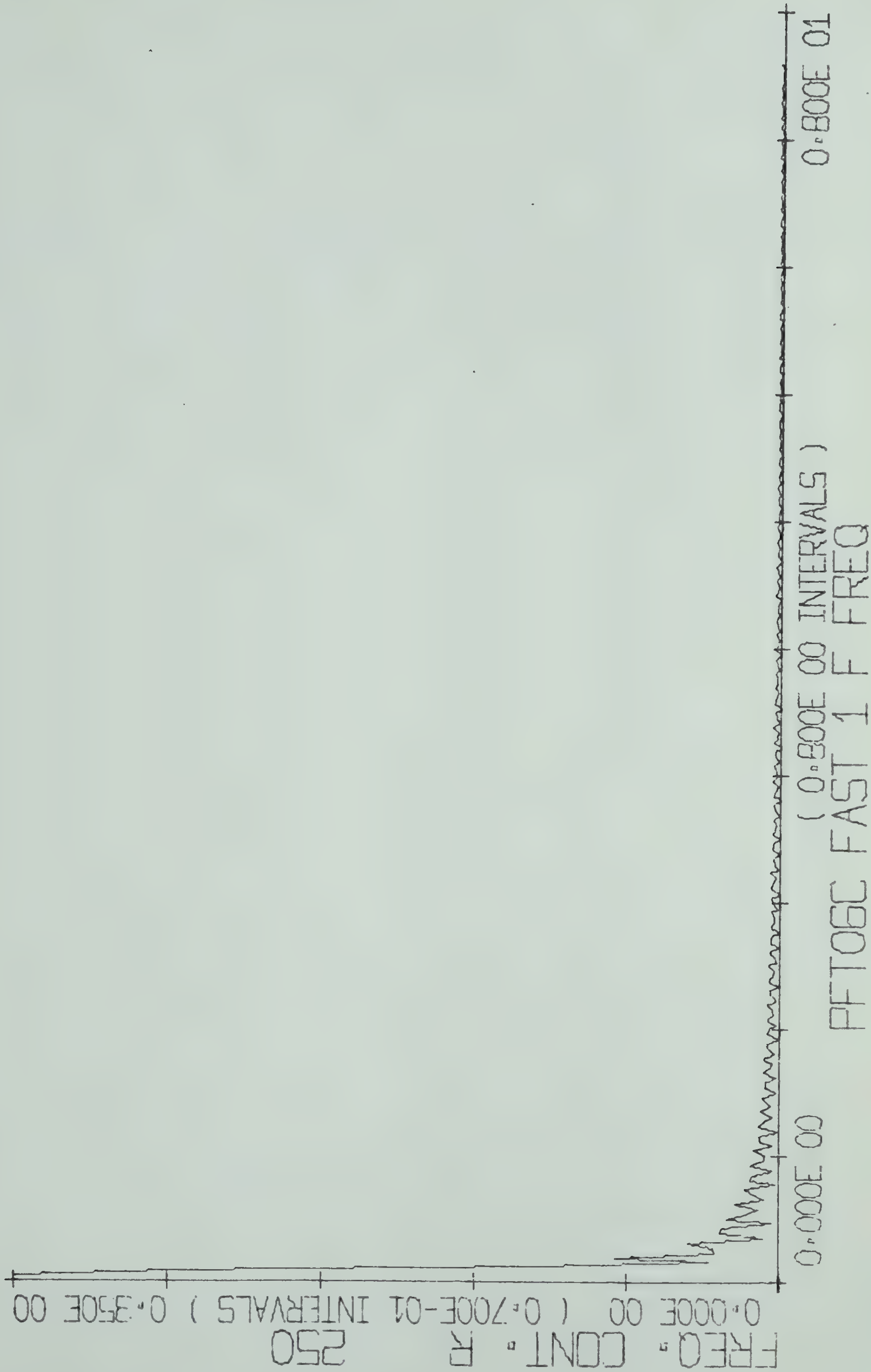

0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
PTHO4B THRY 1 F TIME
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
OUT PULSE R 1200

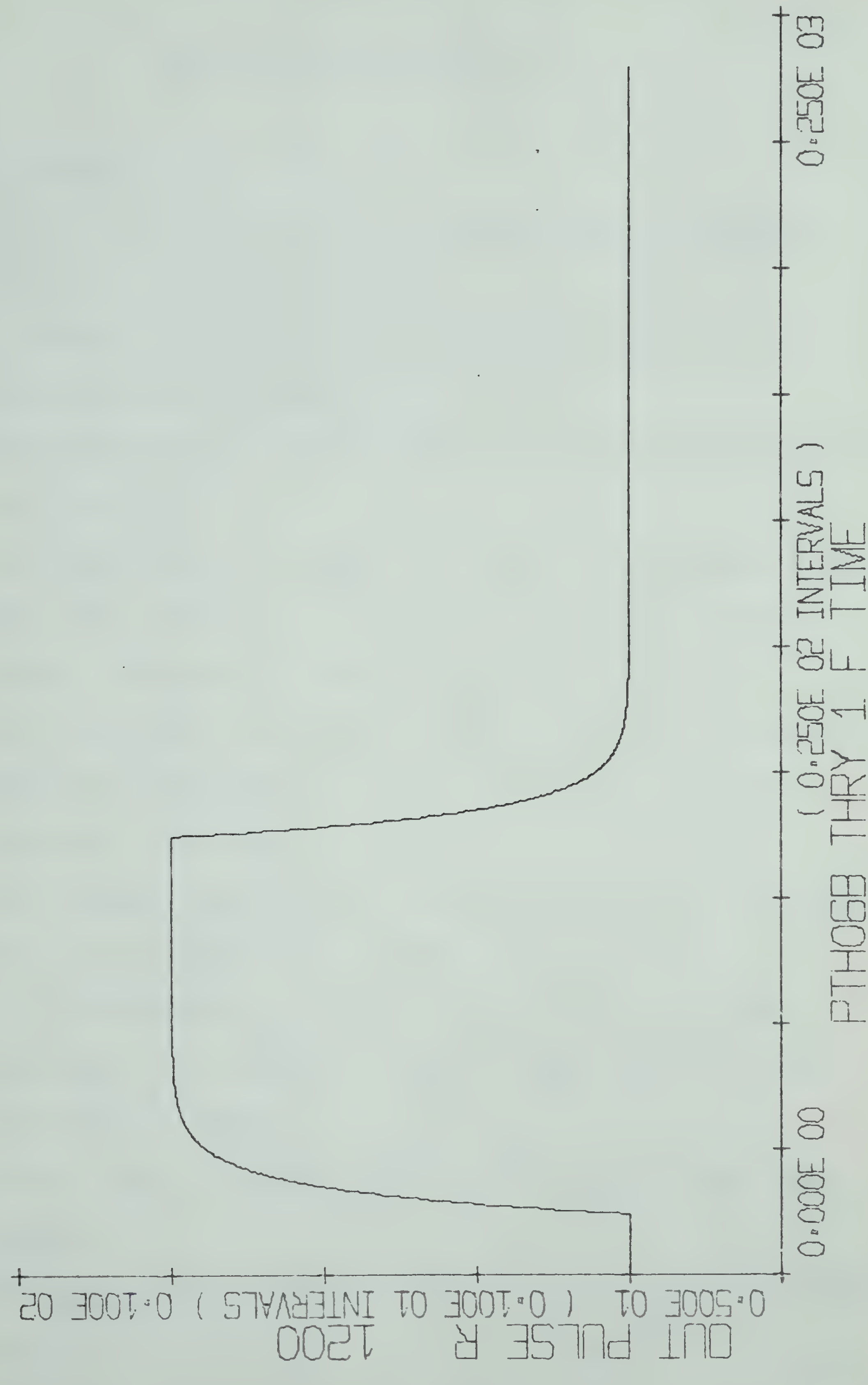

0.000E 00
(0.150E 02 INTERVALS)
0.150E 03
PTHOSA THRY 1 F TIME
0.500E 01
(0.100E 01 INTERVALS)
0.100E 02
INPUT PULSE R


PFT05C FAST 1 F FREQ
(0.800E 00 INTERVALS)
0.000E 00
0.800E 01
FREQ. CONT. R 125
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01


0.000E 00
0.150E 03
(0.150E 02 INTERVALS)
PTH05B THRY 1 F TIME
0.500E 01
(0.100E 01 INTERVALS)
0.100E 02
OUT PULSE R 1200


PTHOGA THRY 1 F TIME
(0.250E 02 INTERVALS)
0.000E 00
0.250E 03
INPUT PULSE R 1200
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02


0.000E 00
0.800E 01
(0.800E 00 INTERVALS)
PFTOGC FAST 1 F FREQ
FREQ. CONT. R 250
0.000E 00 (0.700E-01 INTERVALS) 0.350E 00


OUT PULSE R 1200
0.500E 01 (0.100E 01 INTERVALS) 0.100E 02
0.000E 00
0.250E 03
(0.250E 02 INTERVALS)
PTHO6B THRY 1 F TIME

APPENDIX II

EXPERIMENTAL PULSE GRAPHS

General Comments

a. Table II A describes the graphs presented in this appendix in the order in which they appear.

b. A description of the code used for labelling the axes of the graphs was given in Section 6.1.

c. The data presented in these graphs was obtained using the techniques outlined in Chapter V.

d. All input pulses are a measure of the test heat exchanger tube flow when pulsed from a steady state value of approximately eight to a maximum of twelve U.S. gallons per minute.

e. All output pulses are a measure of the response of the test heat exchanger shell outlet temperature to the tube flow pulses described in (d) above.

f. All frequency content graphs have the same scale on the abscissa but not necessarily on the ordinate.

g. All frequency content curves were calculated using Filon's quadrature except for PFT14C which was determined using the Fast Fourier transform.

h. For all runs, the sample interval for the time data was 0.025 seconds.

i. The plots associated with each run are arranged in the following order:

1. input pulse

2. frequency content of input pulse

3. output pulse

(Note: For run 13 more than one version of the input and output pulse is presented.)

j. The runs are not arranged in the order of run number, but in order of input pulse duration starting with the rectangular shape and following with the ramp shape.

TABLE II A

LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR EXPERIMENTAL RUNS

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	COMMENT
PEX14A	299	1.0	Input pulse; rectangular.
PFN14C	300	1.0	Frequency content curve for input pulse PEX14A.
PFT14C	301	1.0	Frequency content curve for input pulse PEX14A covering a wider frequency range than PFN14C.
PEX14B	302	1.0	Output pulse; response to input pulse PEX14A.
PEX10A	303	3.0	Input pulse; rectangular.
PFN10C	304	3.0	Frequency content curve for input pulse PEX10A.
PEX10B	305	3.0	Output pulse; response to input pulse PEX10A.
PEX12A	306	6.0	Input pulse; rectangular.
PFN12C	307	6.0	Frequency content curve for input pulse PEX12A.
PEX12B	308	6.0	Output pulse; response to input pulse PEX12A.
PEX16A	309	6.0	Input pulse; rectangular.

TABLE II A (CONT'D)

LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR EXPERIMENTAL RUNS

GRAPH IDENTIFICATION	PAGE	INPUT PULSE DURATION (SEC)	COMMENT
PFN16C	310	6.0	Frequency content curve for input pulse PEX16A.
PEX16B	311	6.0	Output pulse response to input pulse PEX16A.
PEX11A	312	10.0	Input pulse; rectangular.
PFN11C	313	10.0	Frequency content curve for input pulse PEX11A.
PEX11B	314	10.0	Output pulse; response to input pulse PEX11A.
PEX17A	315	15.0	Input pulse; rectangular.
PFN17C	316	15.0	Frequency content curve for input pulse PEX17A.
PEX17B	317	15.0	Output pulse; response to input pulse PEX17A.
PEX18A	318	6.0	Input pulse; ramp.
PFN18C	319	6.0	Frequency content curve for input pulse PEX18A.
PEX18B	320	6.0	Output pulse; response to input pulse PEX18A.

TABLE II A (CONT'D)

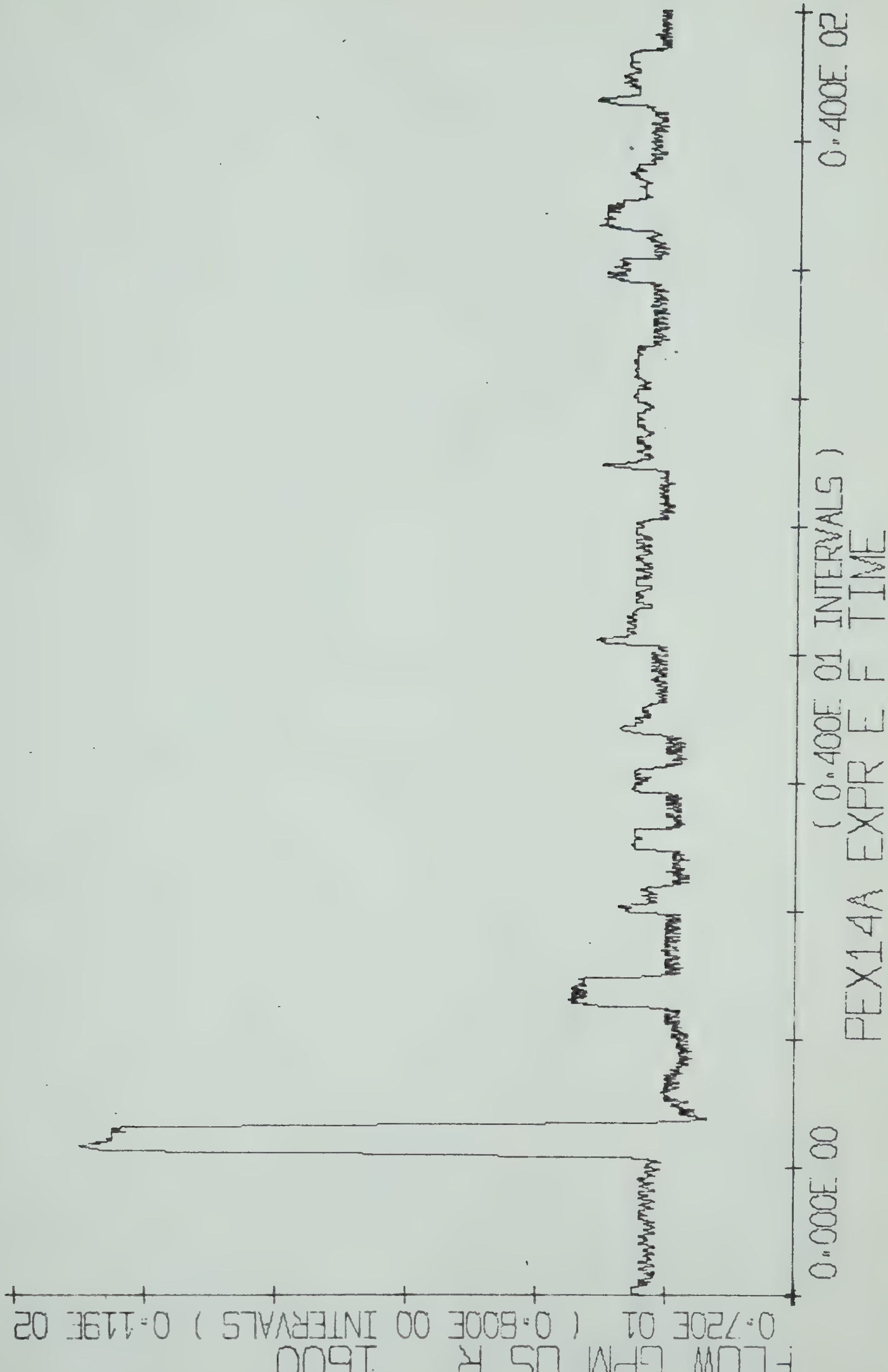
LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR EXPERIMENTAL RUNS

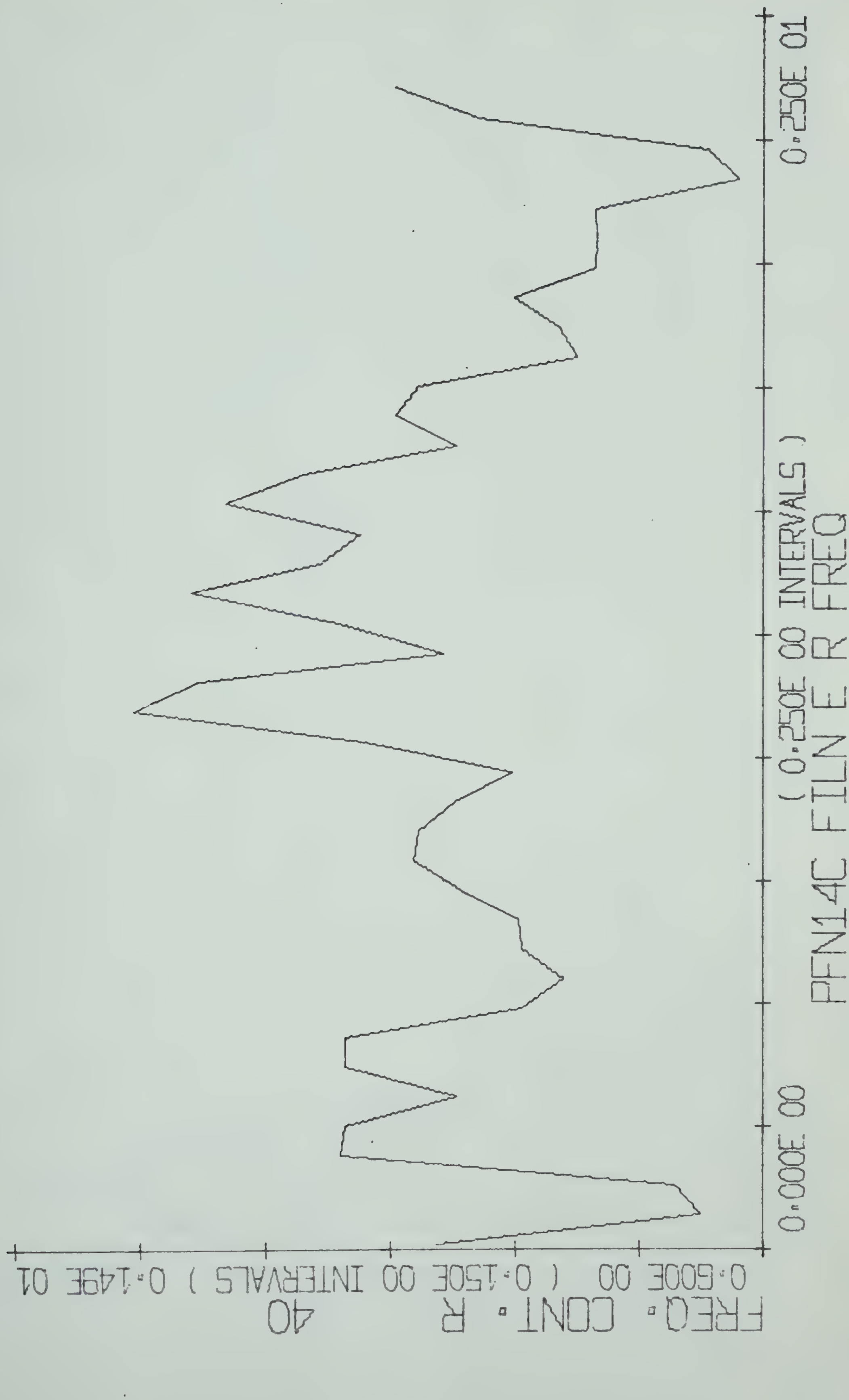
GRAPH IDENTIFICATION	PAGE	INLET PULSE DURATION (SEC)	COMMENT
PEX13A	321	10.0	Input pulse; ramp.
PEX13I	322	10.0	The same data as plotted PEX13A but after third order polynomial smoothing has been applied to it three times (Section 5.4.2.2).
PEX13C	323	10.0	The same data as plotted in PEX13A but after it was reduced using but subroutine REDPT (Section 5.4.2.3) with selection of every fourth point.
PFN13C	324	10.0	Frequency content curve for input pulse PEX13A.
PEX13B	325	10.0	Output pulse; response to input pulse PEX13A.
PEX13J	326	10.0	The same data as plotted in PEX13B after third order polynomial smoothing using eleven points has been applied to it three times (Section 5.4.2.2).
PEX13D	327	10.0	The same data as plotted in PEX13B after it was reduced using subroutine REDPT (Section 5.4.2.3) with selection of every fourth point.

TABLE II A (CONT'D)

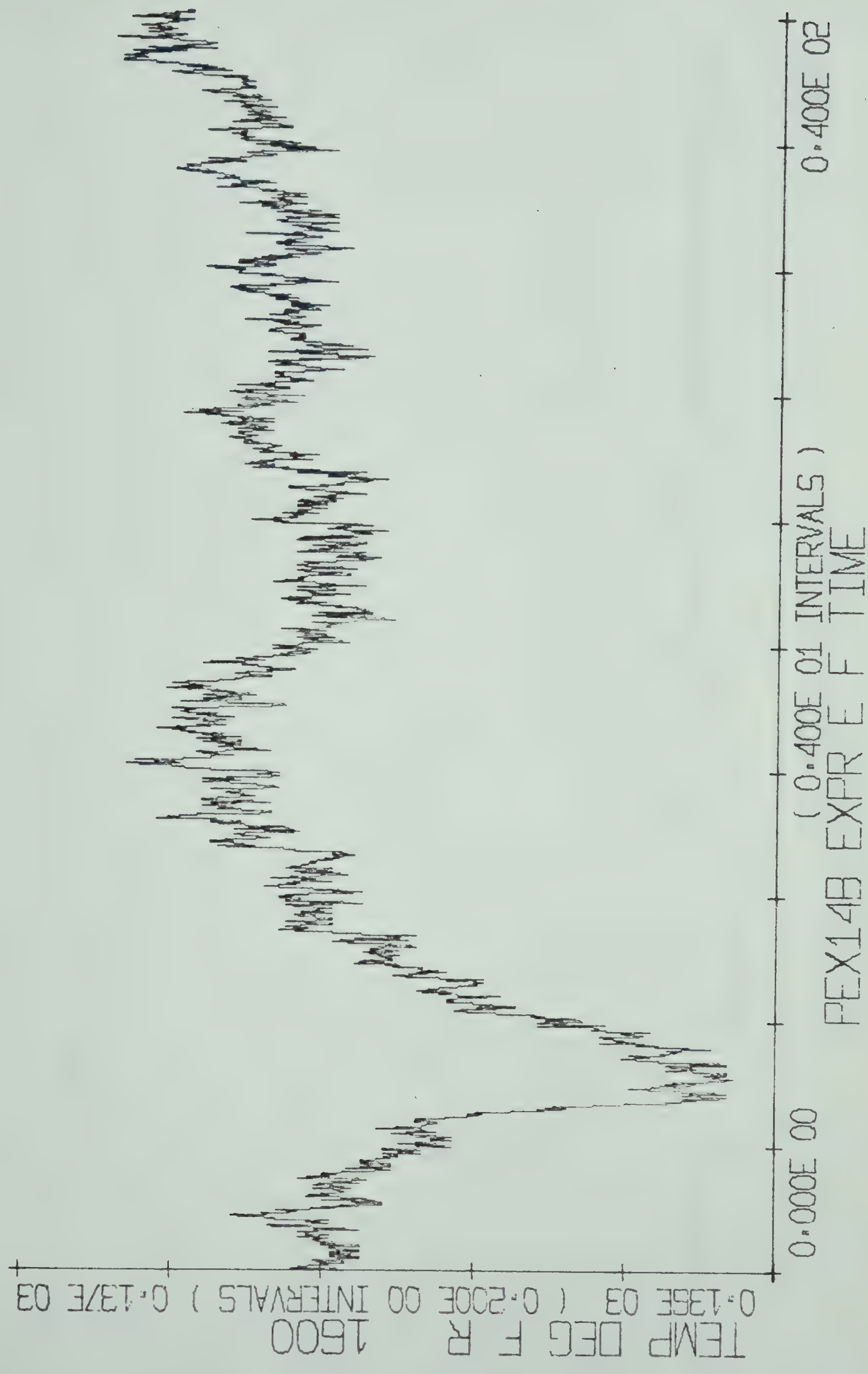
LIST OF GRAPHS OF INPUT PULSES, FREQUENCY CONTENT AND OUTPUT PULSES FOR EXPERIMENTAL RUNS

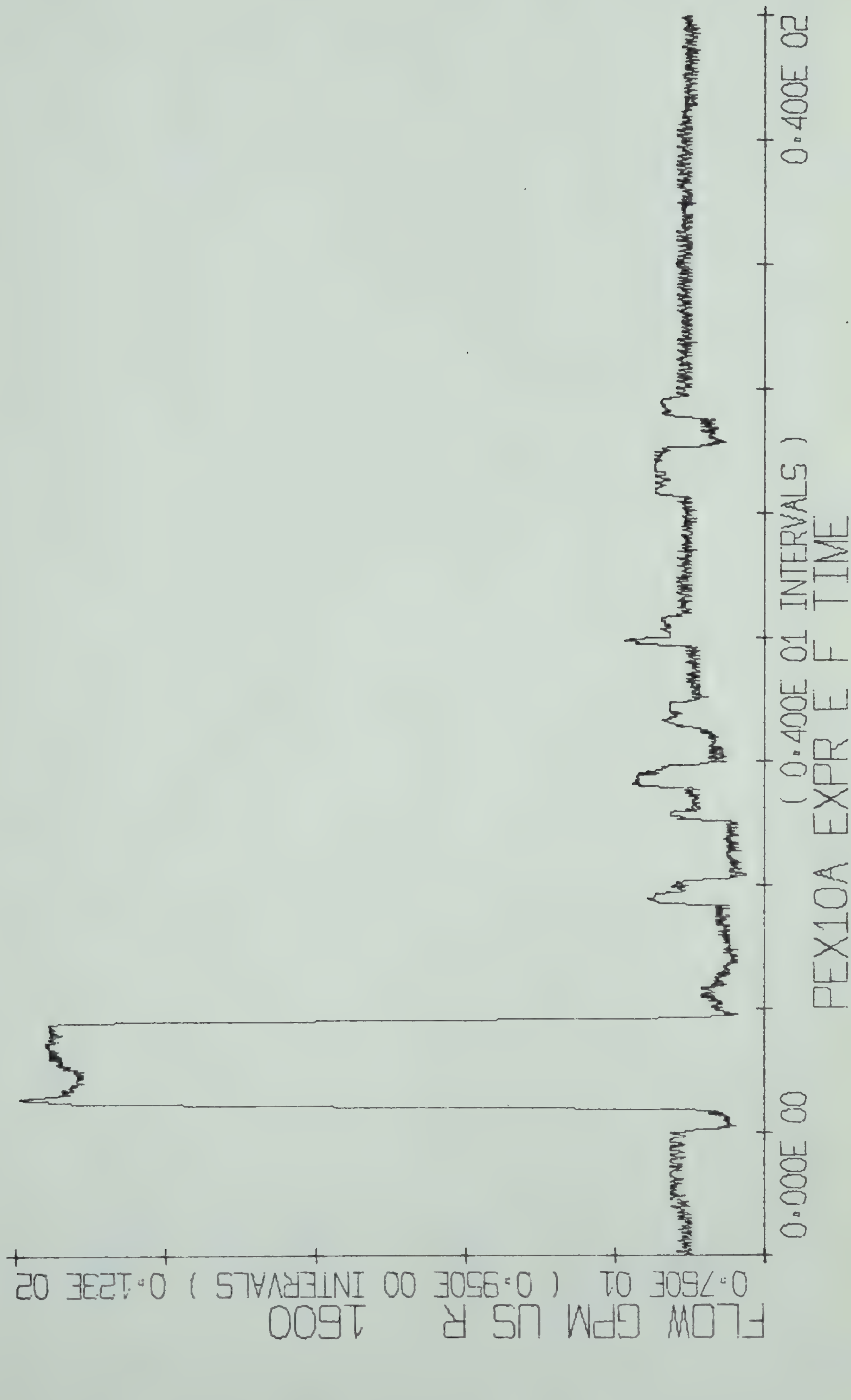
GRAPH IDENTIFICATION	PAGE	INLET PULSE DURATION (SEC)	COMMENT
PEX19A	328	15.0	Input pulse; ramp.
PFN19C	329	15.0	Frequency content curve for input pulse PEX19A.
PEX19B	330	15.0	Output pulse; response to input pulse PEX19A.
PEX20A	331	20.0	Input pulse; ramp.
PFN20C	332	20.0	Frequency content curve for input pulse PEX20A.
PFN20B	333	15.0	Output pulse; response to input pulse PEX20A.


0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX14A EXPR E F TIME
0.720E 01 (0.800E 00 INTERVALS) 0.119E 02
FLOW GPM US R 1600

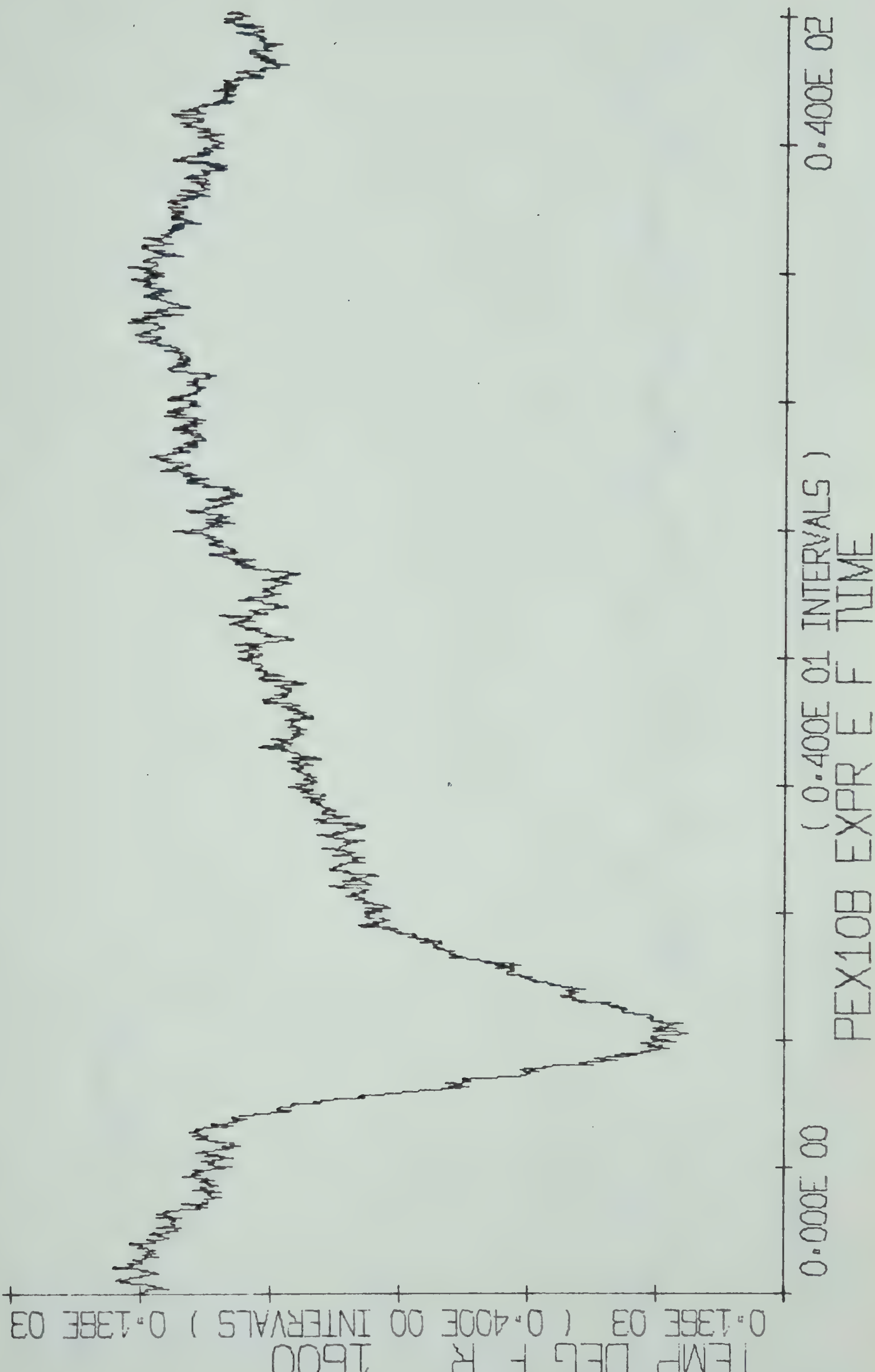

0.250E 01
(0.250E 00 INTERVALS)
PFN14C FILN E R FREQ
0.000E 00
0.600E 00 (0.150E 00 INTERVALS) 0.149E 01
FREQ. CONT. R
40

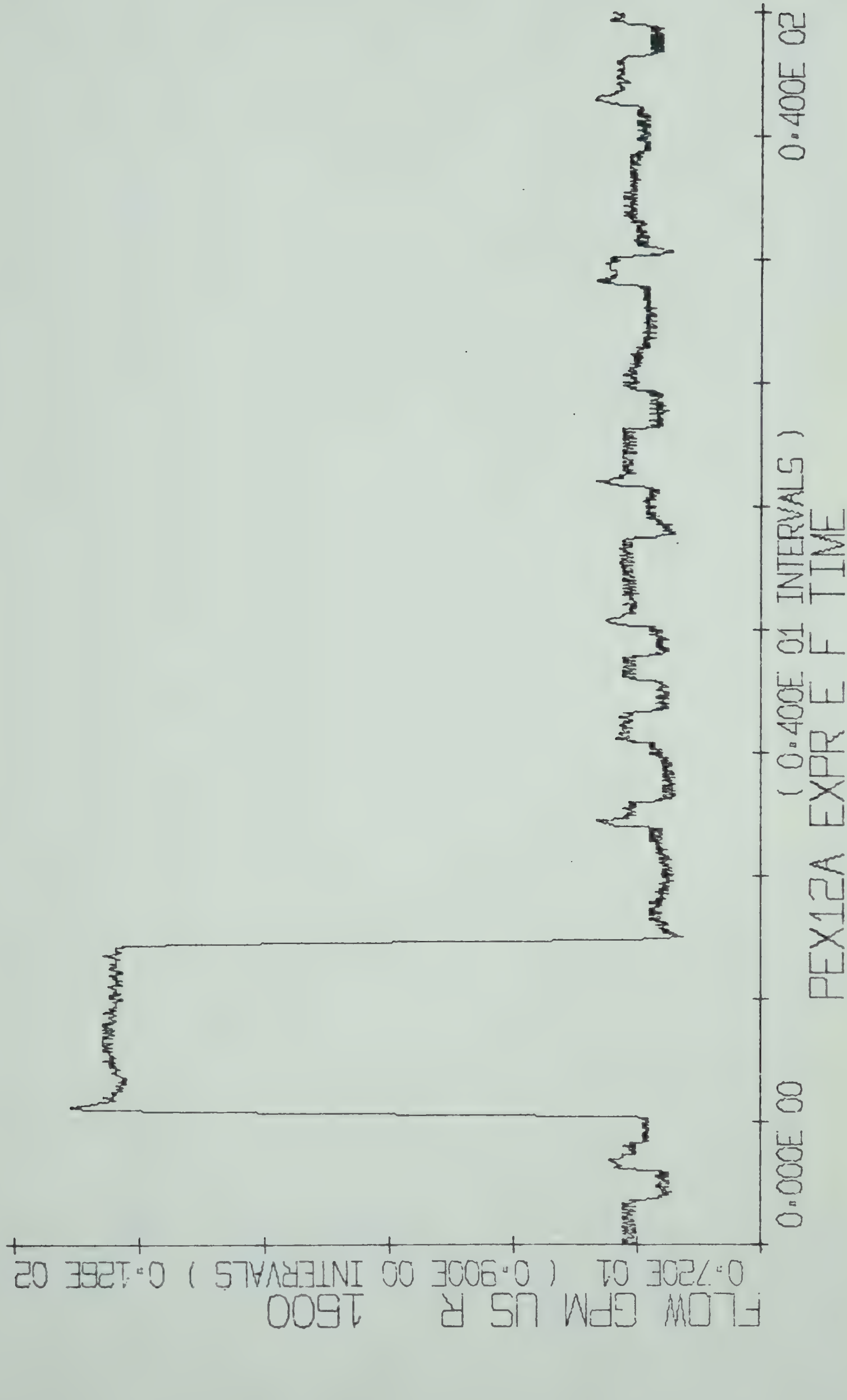

0.000E 00
0.350E 02
(0.350E 01 INTERVALS)
PFT14C FAST E F FREQ
0.000E 00 (0.300E 01 INTERVALS) 0.150E 02
FREQ. CONT. R 125

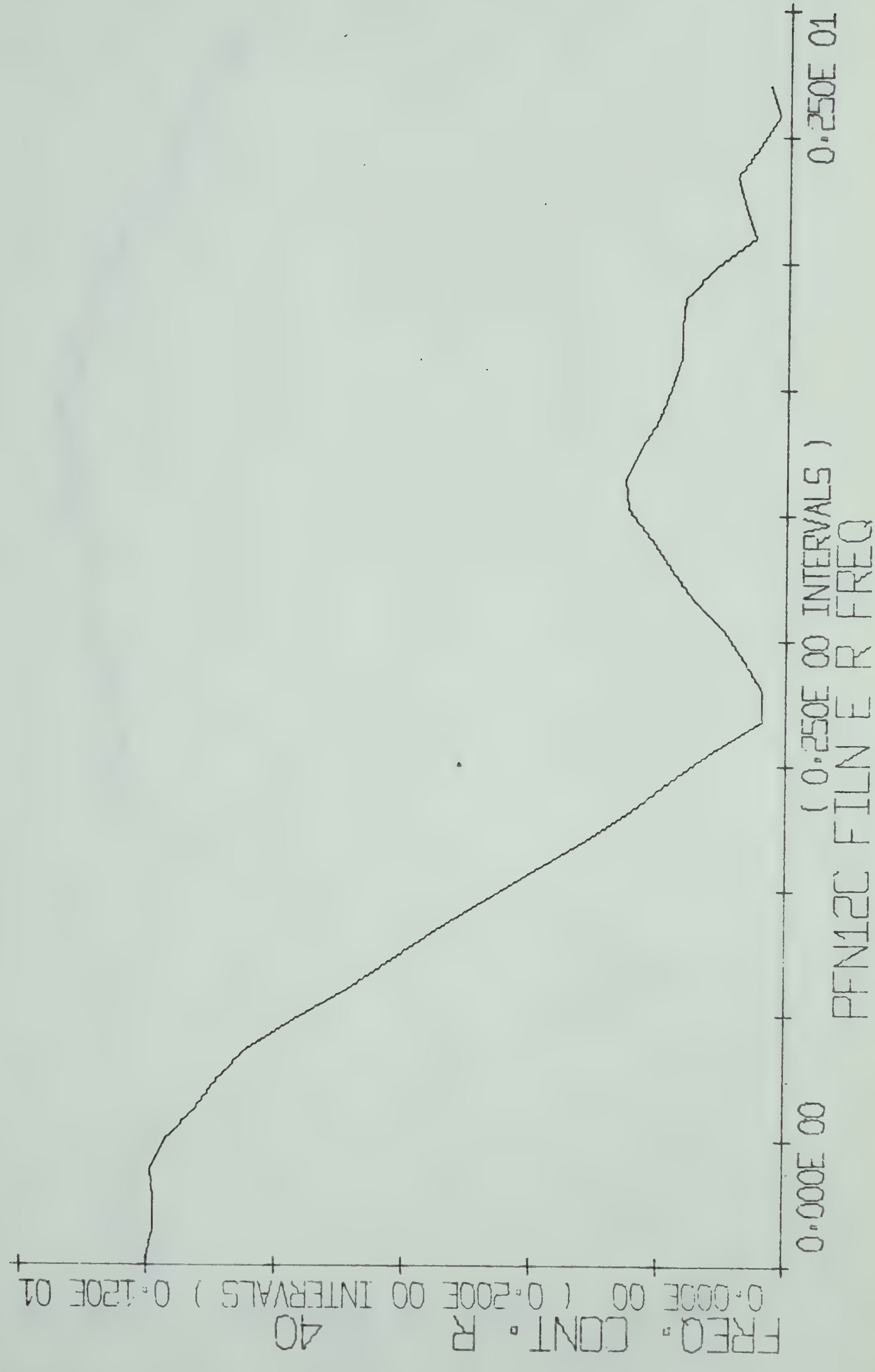

0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX14B EXPR E F TIME
0.135E 03 (0.200E 00 INTERVALS) 0.137E 03
TEMP DEG F R 1600

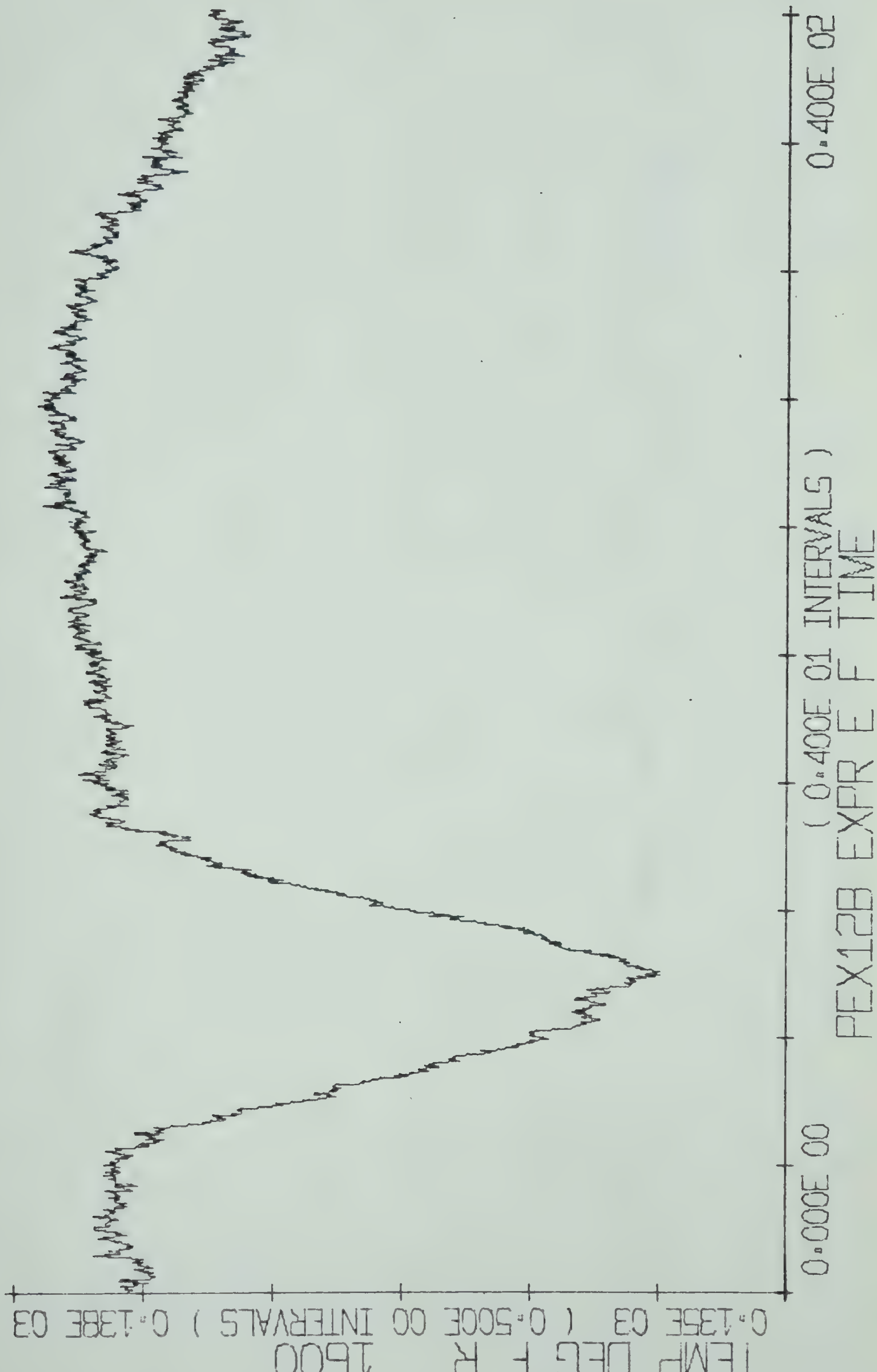

0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX10A EXPR E F TIME
0.760E 01 (0.950E 00 INTERVALS) 0.123E 02
FLOW GPM US R 1600


0.250E 01
(0.250E 00 INTERVALS)
PFN10C FILN E R FREQ
0.000E 00
FREQ. CONT. R
40
0.000E 00 (0.250E 00 INTERVALS) 0.125E 01

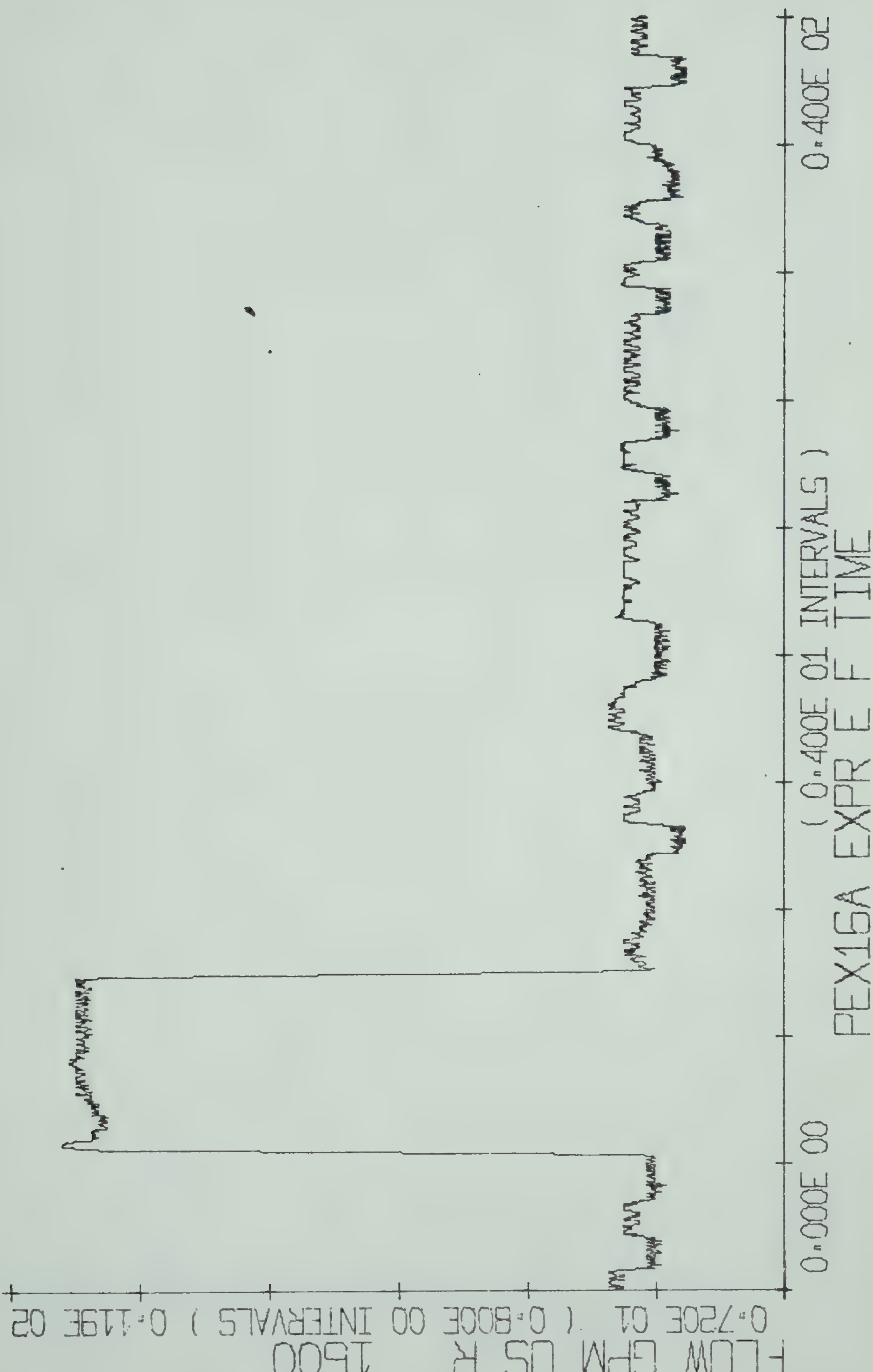

0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX10B EXPR E F TIME
0.138E 03 (0.400E 00 INTERVALS) 0.138E 03
TEMP DEG F R 1600

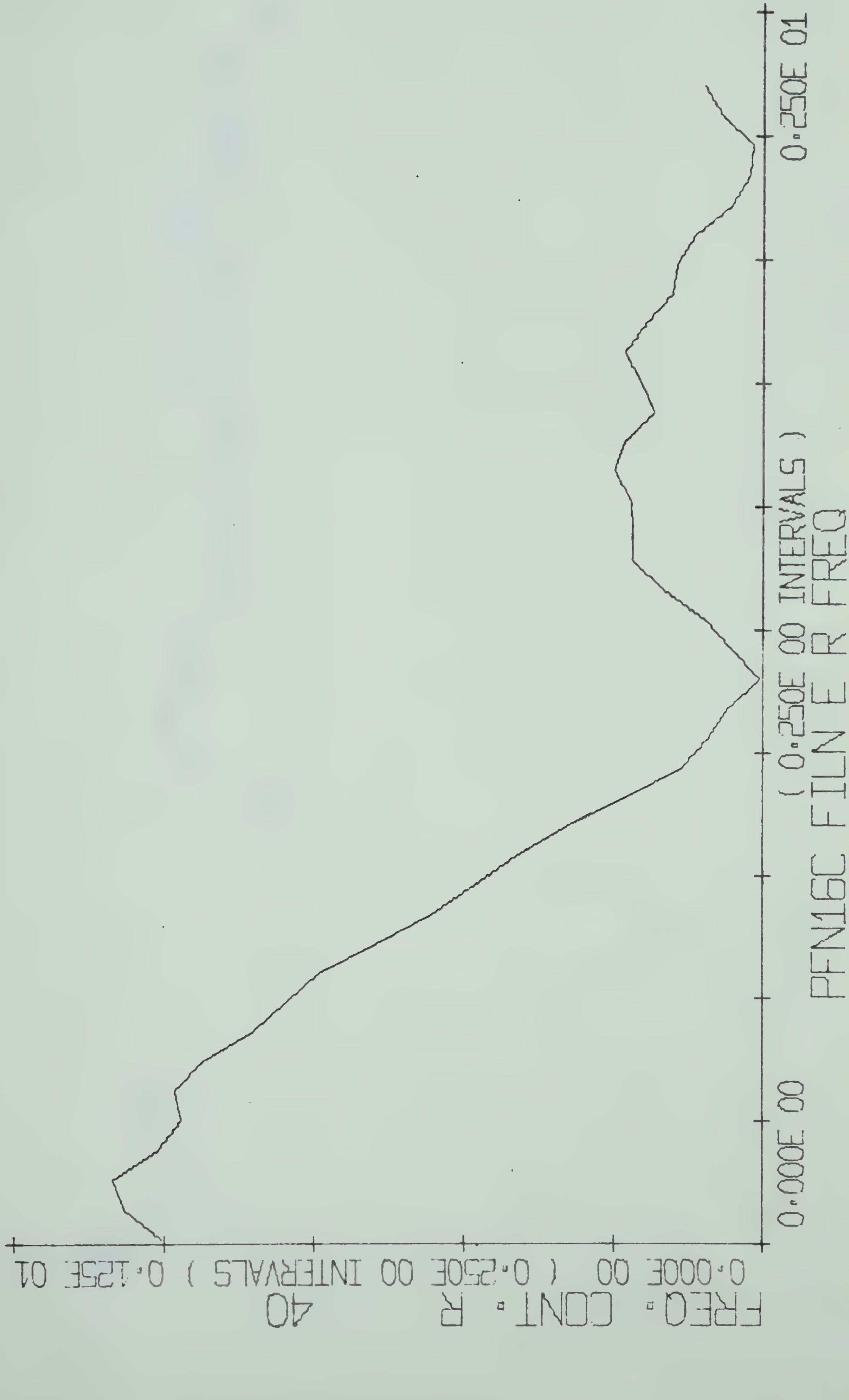

FLOW GPM US R 1500
0.720E 01 (0.900E 00 INTERVALS) 0.126E 02
PEX12A EXPR E F TIME
0.000E 00
(0.400E 01 INTERVALS)
0.400E 02

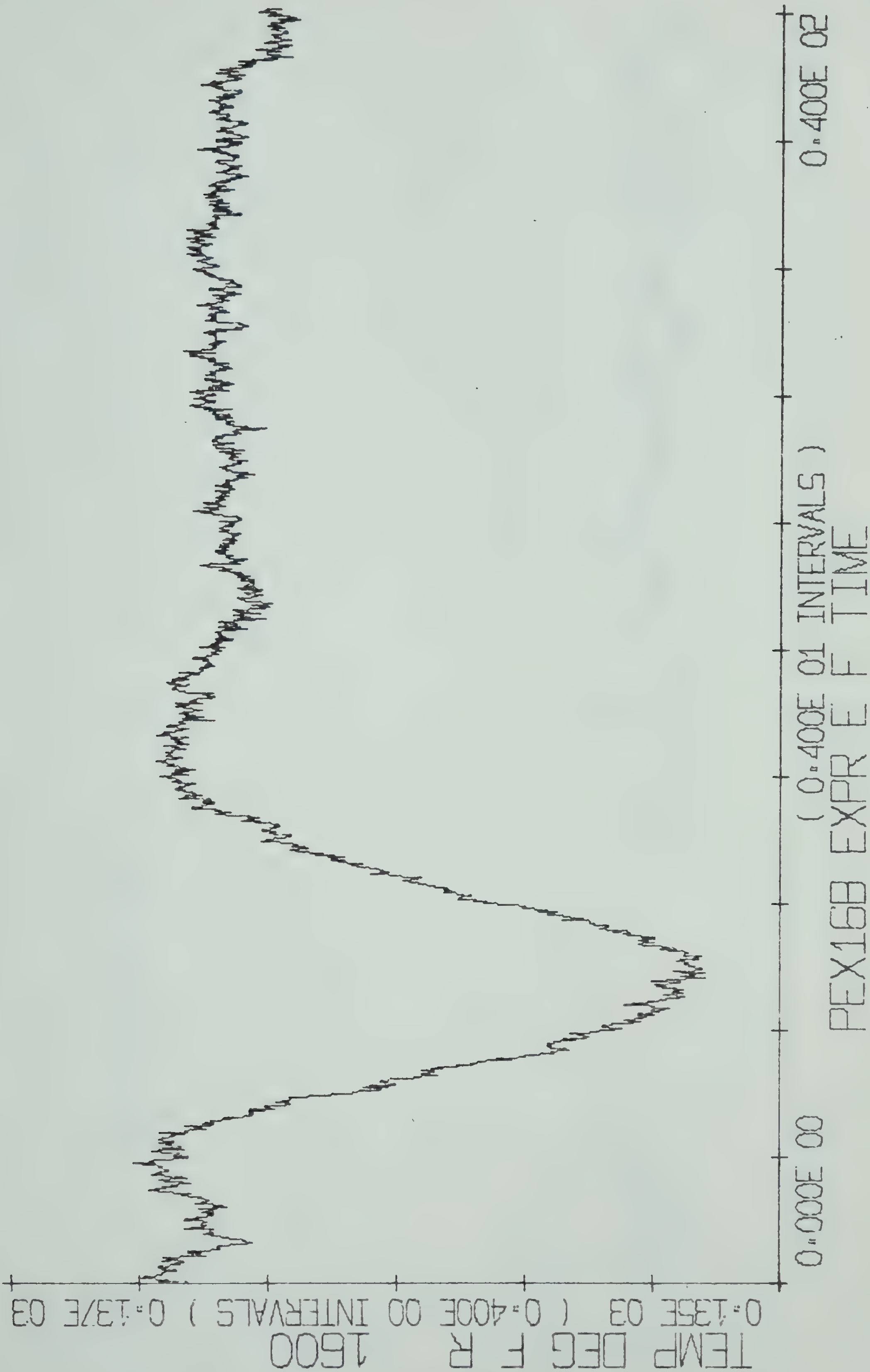

0.000E 00
0.250E 01
(0.250E 00 INTERVALS)
PFN12C FILN E R FREQ
0.000E 00 (0.200E 00 INTERVALS) 0.120E 01
FREQ. CONT. R
40



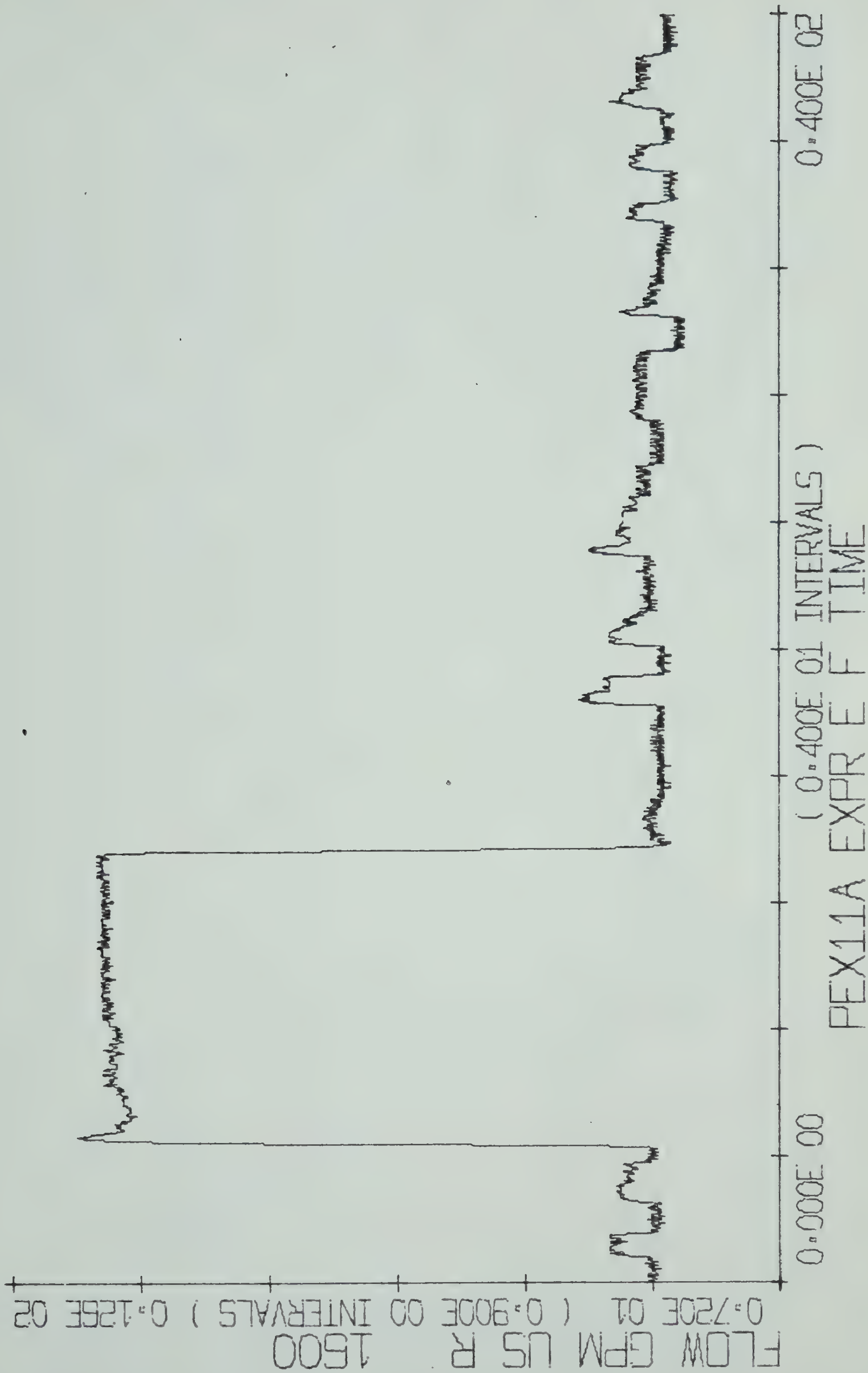
PEX12B EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02
TEMP DEG F R 1600
0.135E 03 (0.500E 00 INTERVALS) 0.138E 03

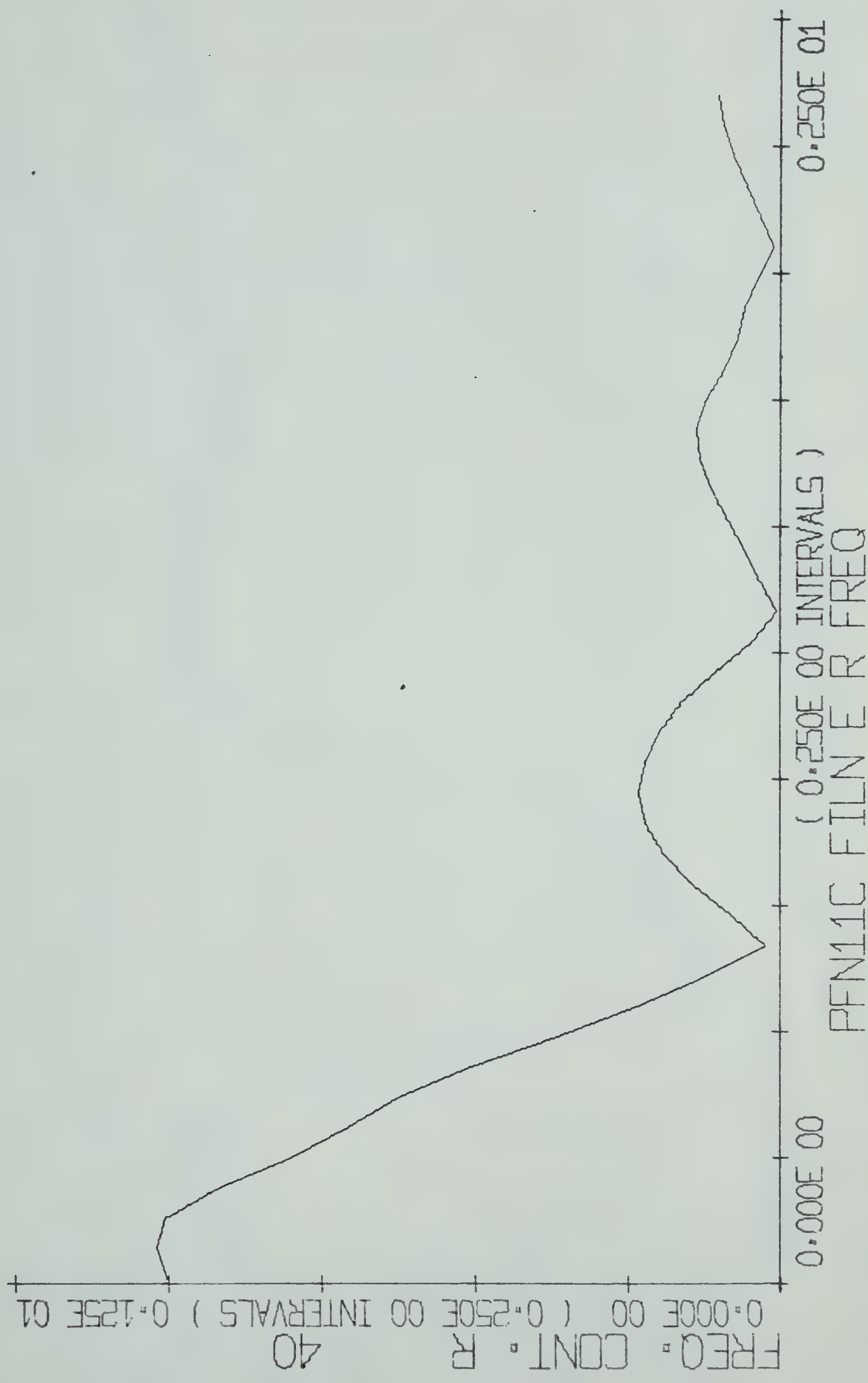

FLOW GPM US R 1500
0.720E 01 (0.800E 00 INTERVALS) 0.119E 02
0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
PEX16A EXPR E F TIME

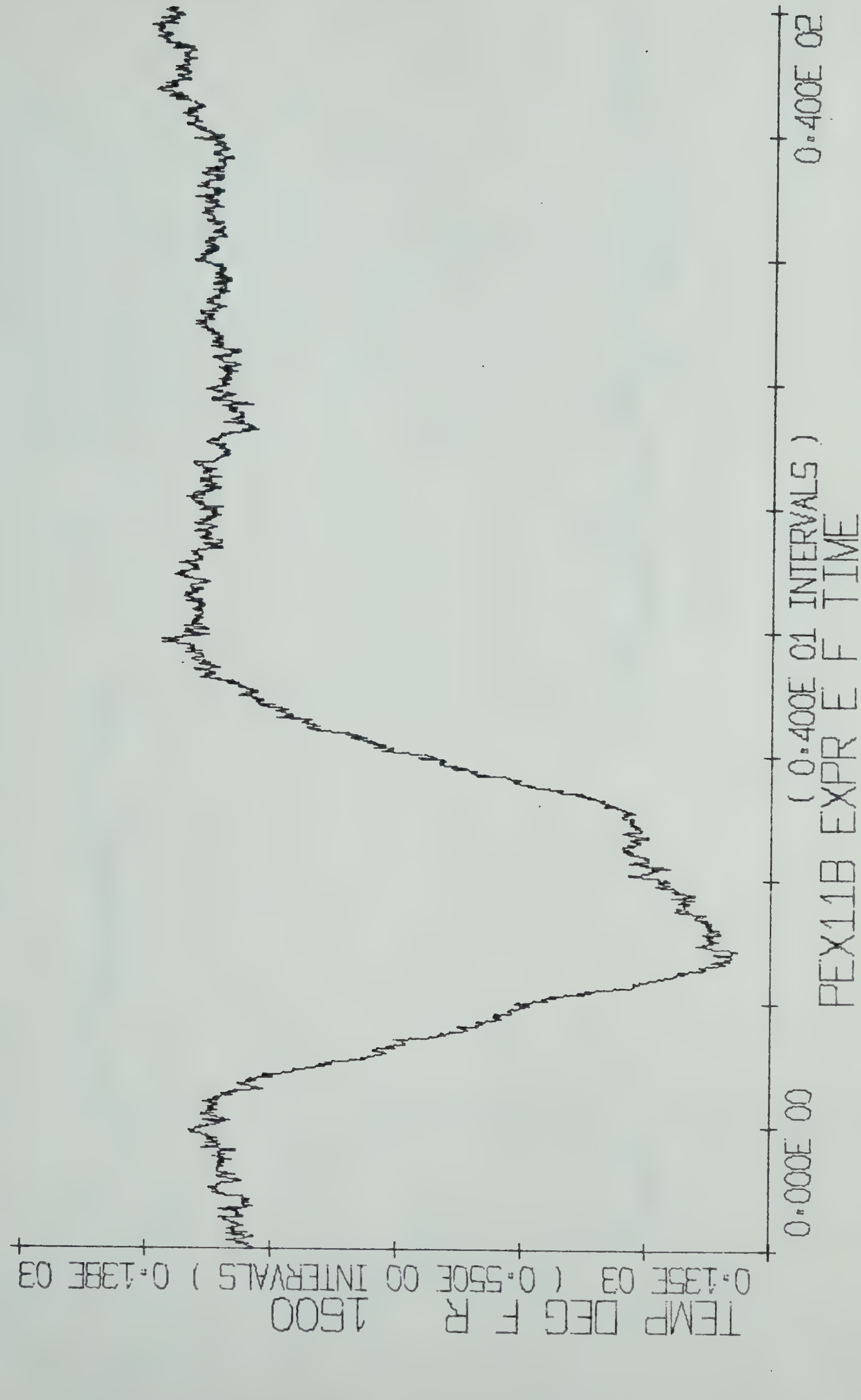

0.000E 00
0.250E 01
(0.250E 00 INTERVALS)
PFN16C FILN E R FREQ
0.000E 00 (0.250E 00 INTERVALS) 0.125E 01
FREQ. CONT. R
40

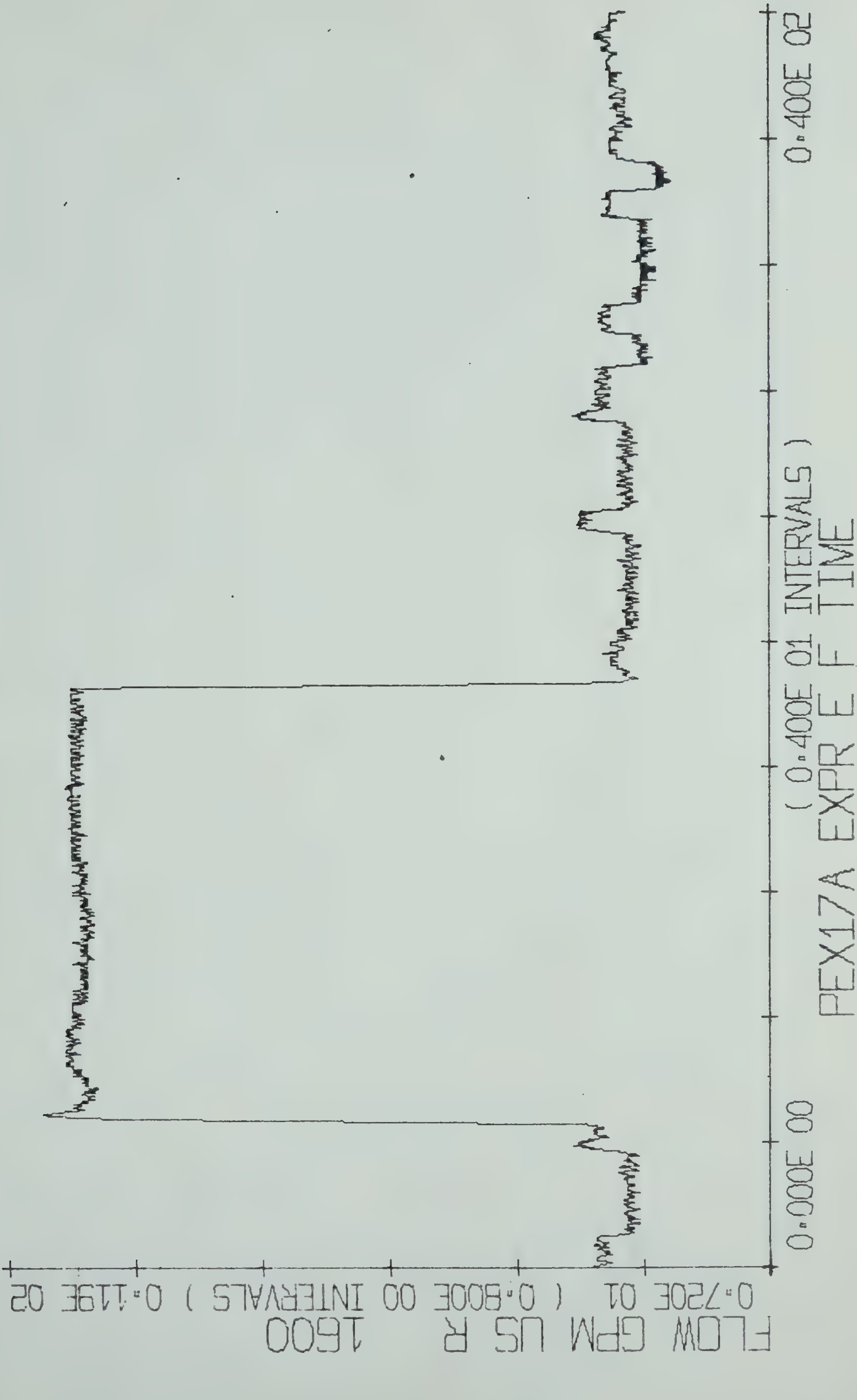


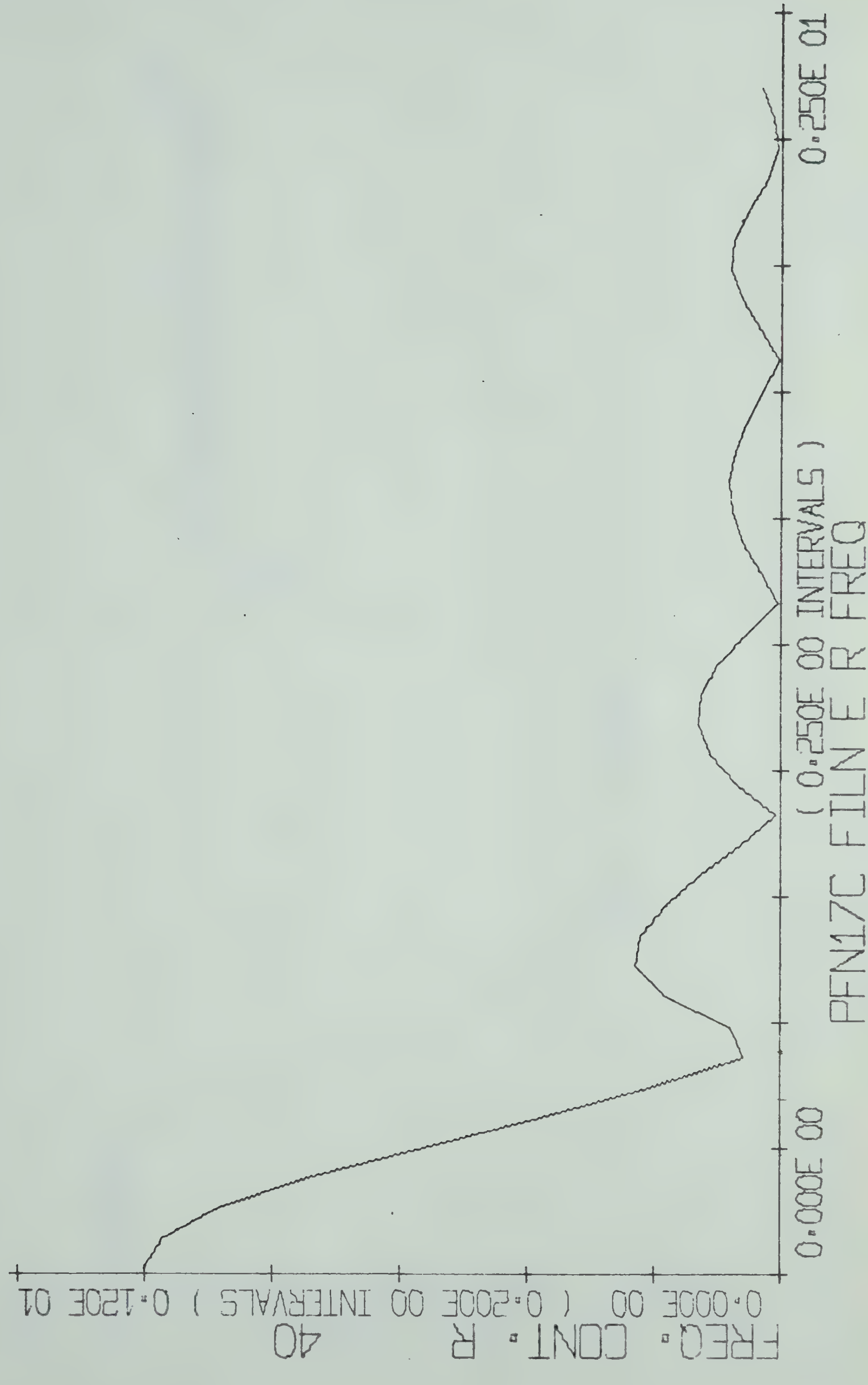
PEX16B EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02
TEMP DEG F R 1600
0.135E 03 (0.400E 00 INTERVALS) 0.137E 03

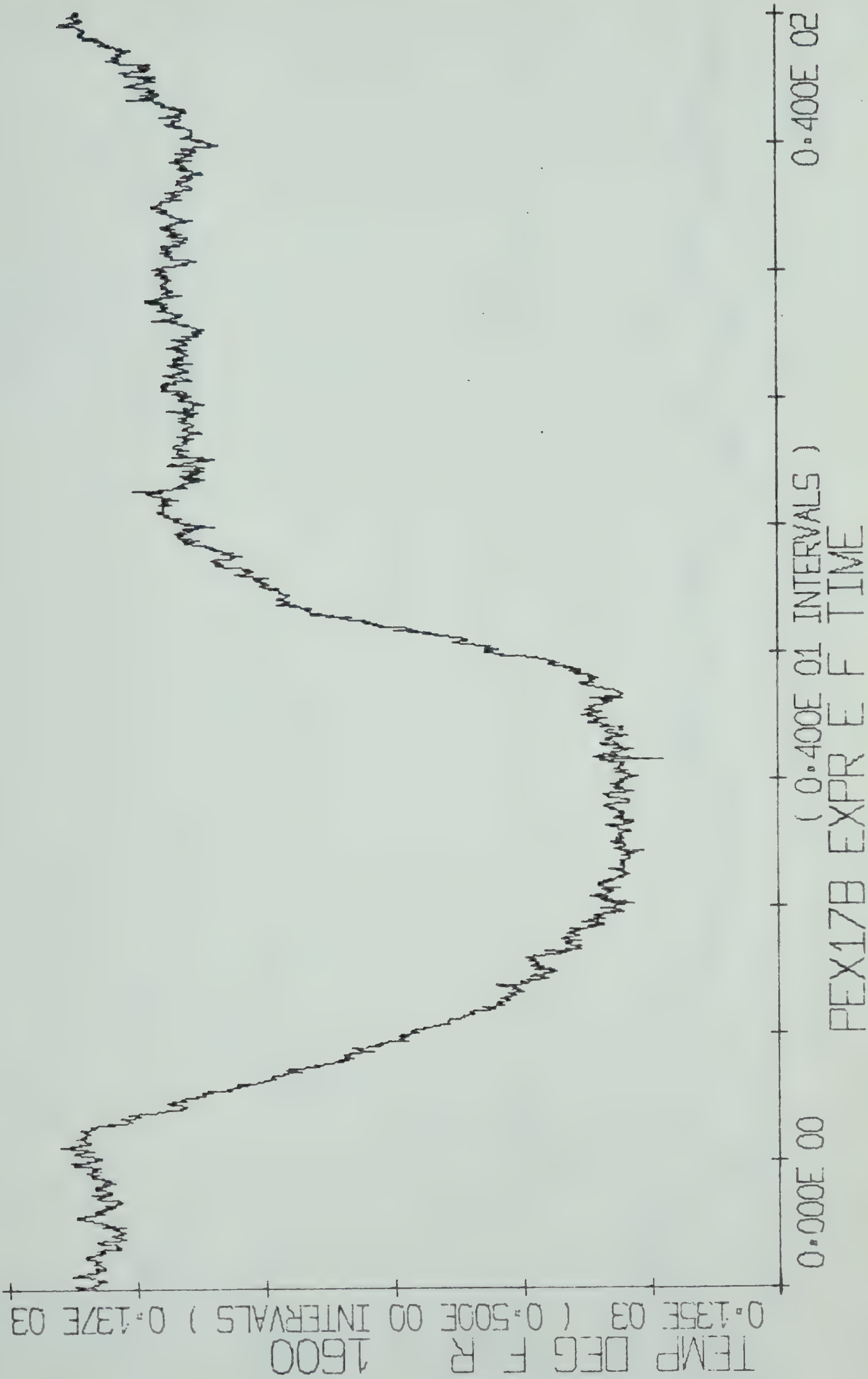

FLOW GPM US R 1500
0.720E 01 (0.900E 00 INTERVALS) 0.125E 02
PEX11A EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02


0.250E 01
0.000E 00
(0.250E 00 INTERVALS)
PFN11C FILN E R FREQ
0.000E 00 (0.250E 00 INTERVALS) 0.125E 01
FREQ. CONT. R
40


TEMP DEG F R 1500
0.135E 03 (0.550E 00 INTERVALS) 0.138E 03
0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
PEX11B EXPR E F TIME

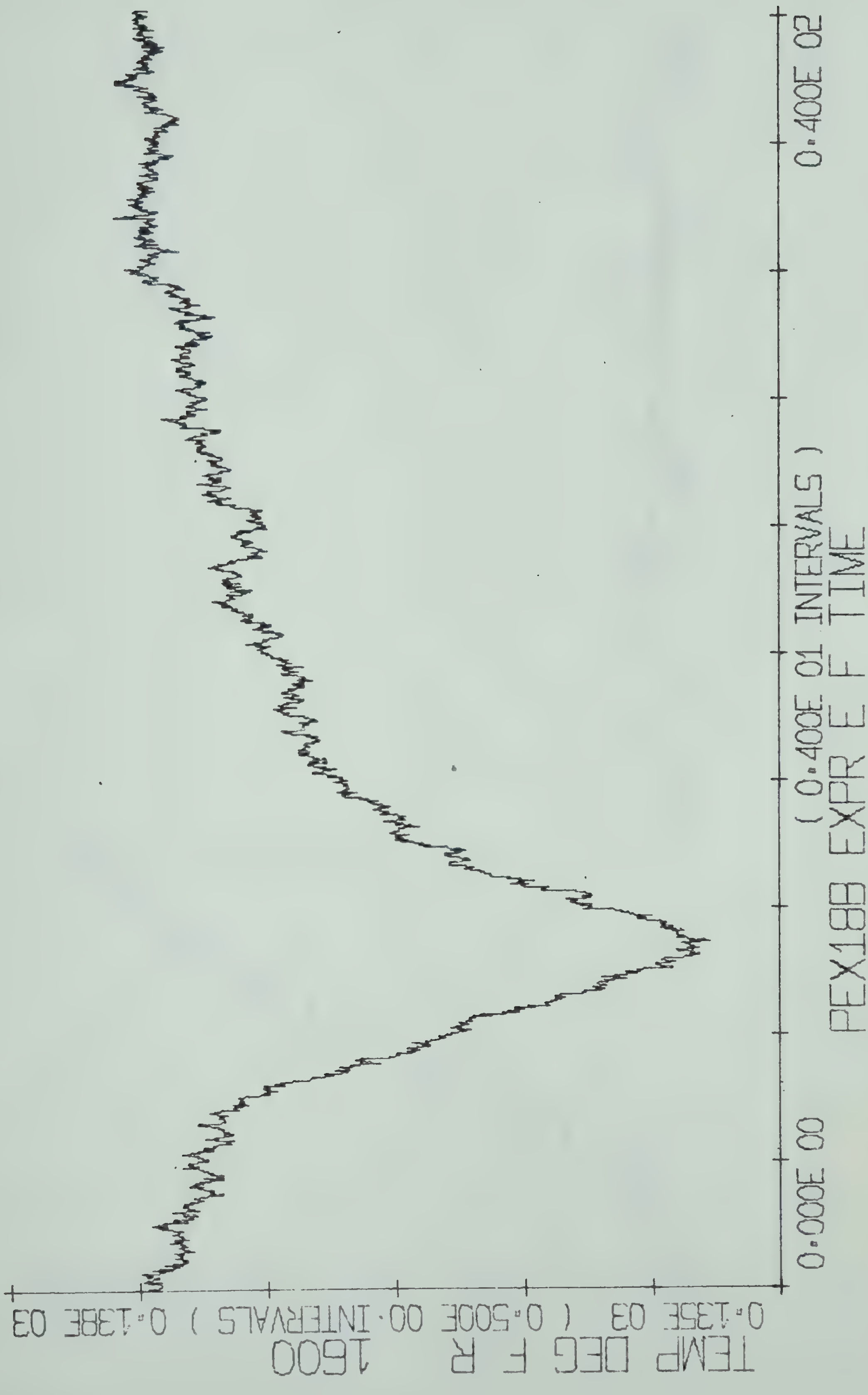

FLOW GPM US R 1600
0.720E 01 (0.800E 00 INTERVALS) 0.119E 02
0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
PEX17A EXPR E F TIME

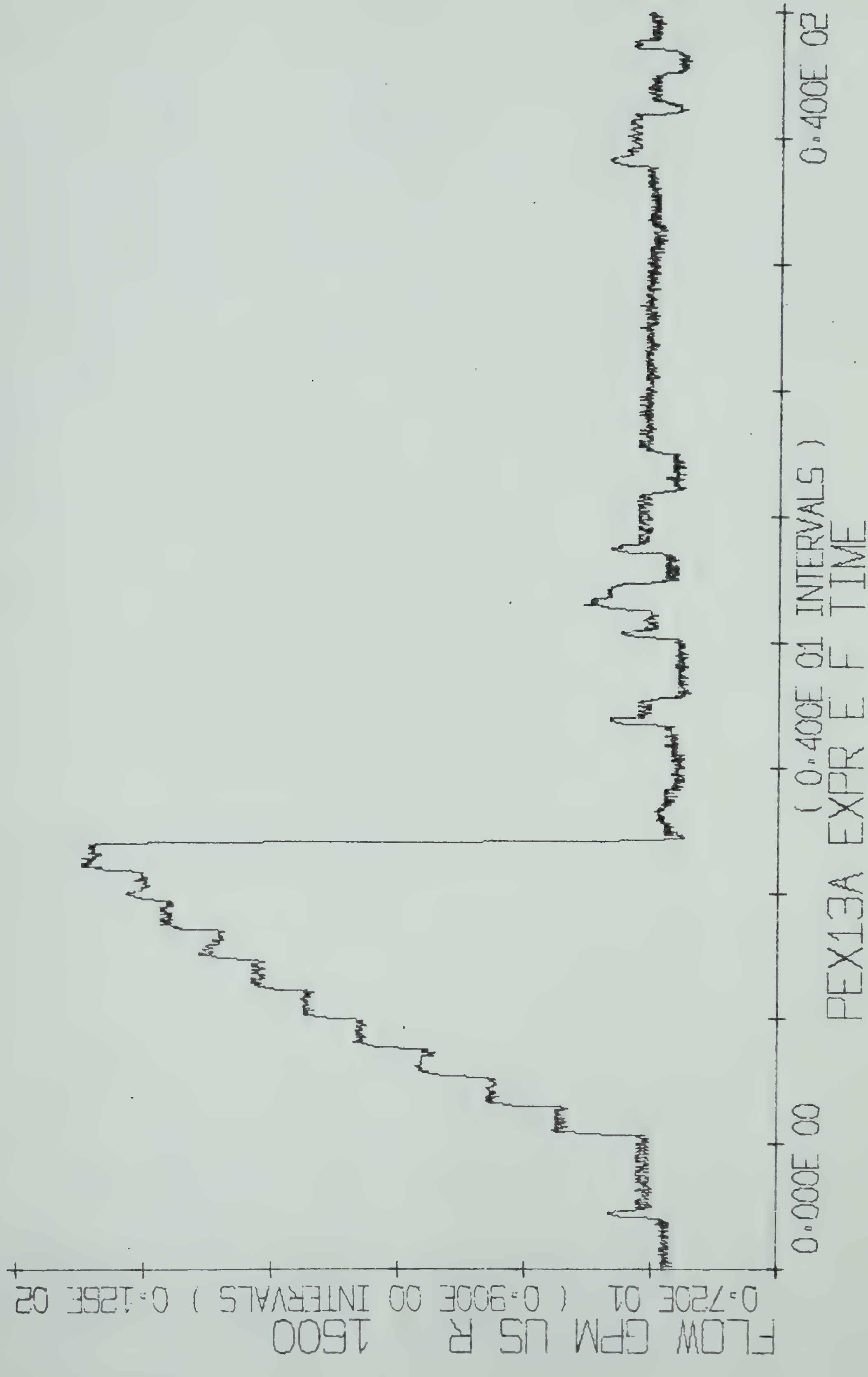

PFN17C FILN E R FREQ
(0.250E 00 INTERVALS)
0.000E 00
0.250E 01
FREQ. CONT. R 40
0.000E 00 (0.200E 00 INTERVALS) 0.120E 01


PEX17B EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02
TEMP DEG F R 1600
0.135E 03 (0.500E 00 INTERVALS) 0.137E 03

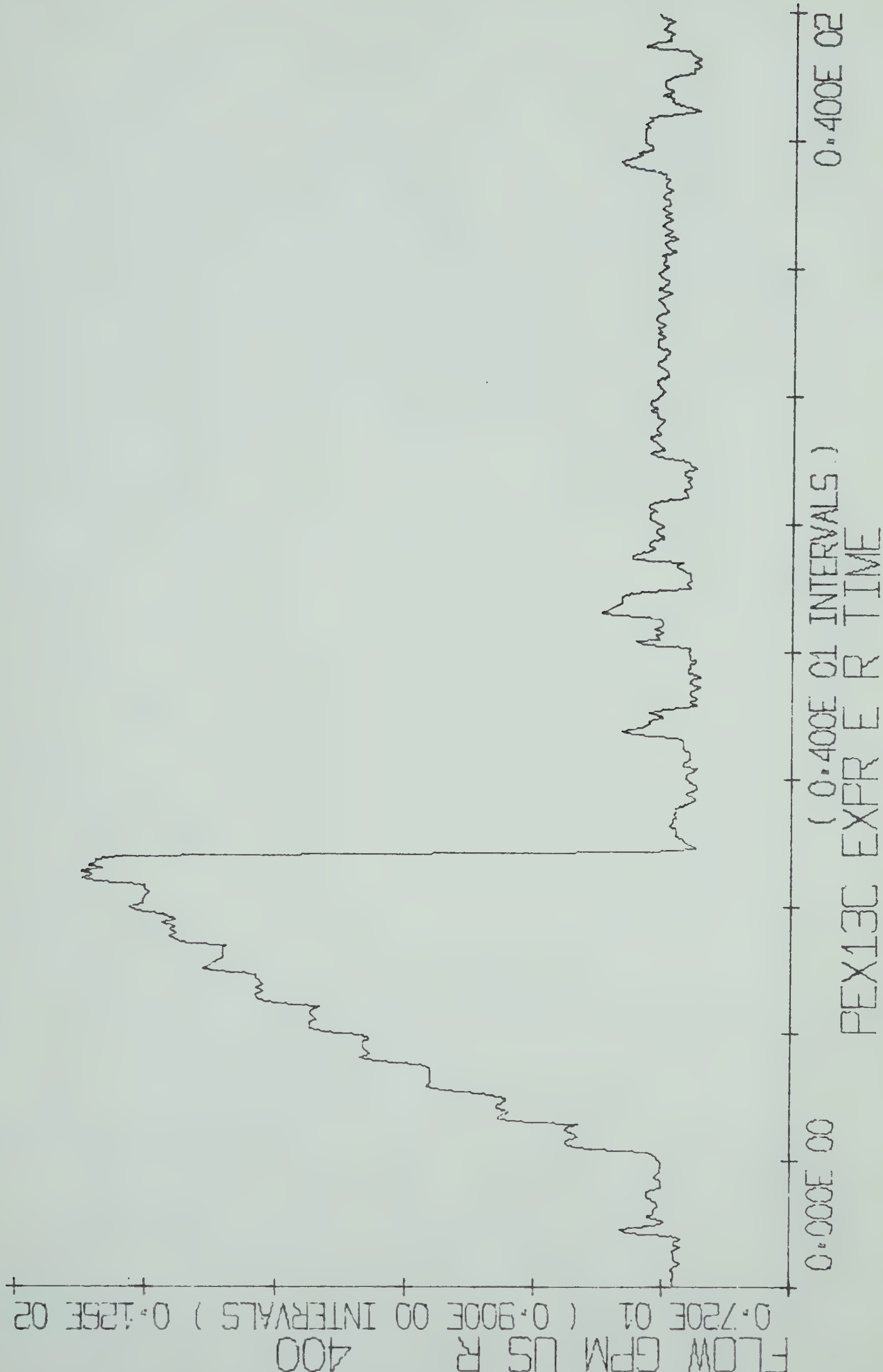

0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX18A EXPR E F TIME
0.765E 01 (0.850E 00 INTERVALS) 0.127E 02
FLOW GPM US R 1600


0.250E 01
(0.250E 00 INTERVALS)
PFN18C FILN E R FREQ
0.000E 00
0.000E 00 (0.300E 00 INTERVALS) 0.180E 01
FREQ. CONT. R
40


0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX18B EXPR E F TIME
0.135E 03 (0.500E 00 INTERVALS) 0.138E 03
TEMP DEG F R 1600

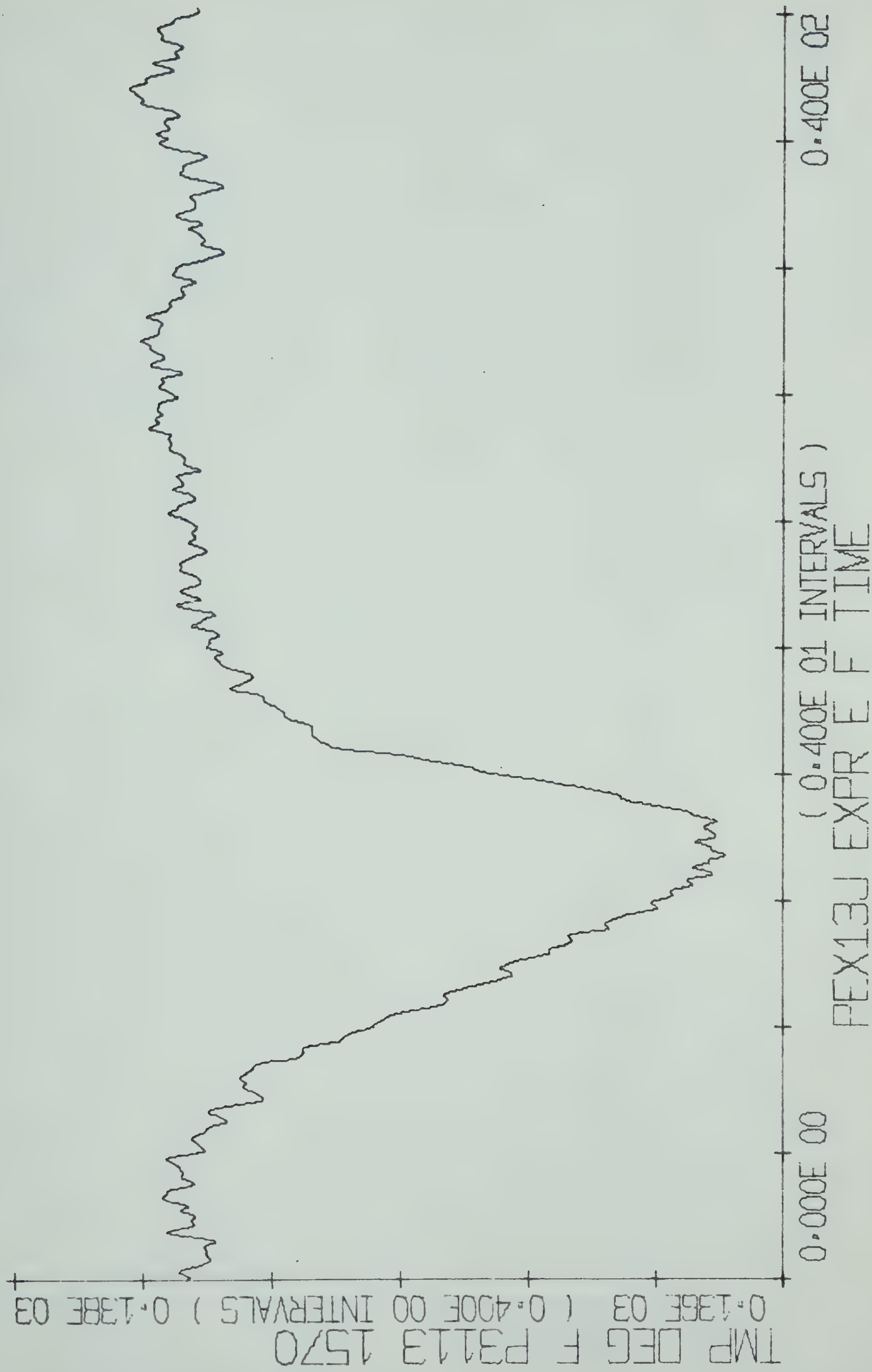

PEX13A EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02
FLOW GPM US R 1500
0.720E 01 (0.900E 00 INTERVALS) 0.126E 02


FLW GPMUS P3113 1570
0.720E 01 (0.900E 00 INTERVALS) 0.126E 02
PEX13I EXPR E F TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02


FLOW GPM US R 400
0.720E 01 (0.900E 00 INTERVALS) 0.125E 02
PEX13C EXPR E R TIME
(0.400E 01 INTERVALS)
0.000E 00
0.400E 02

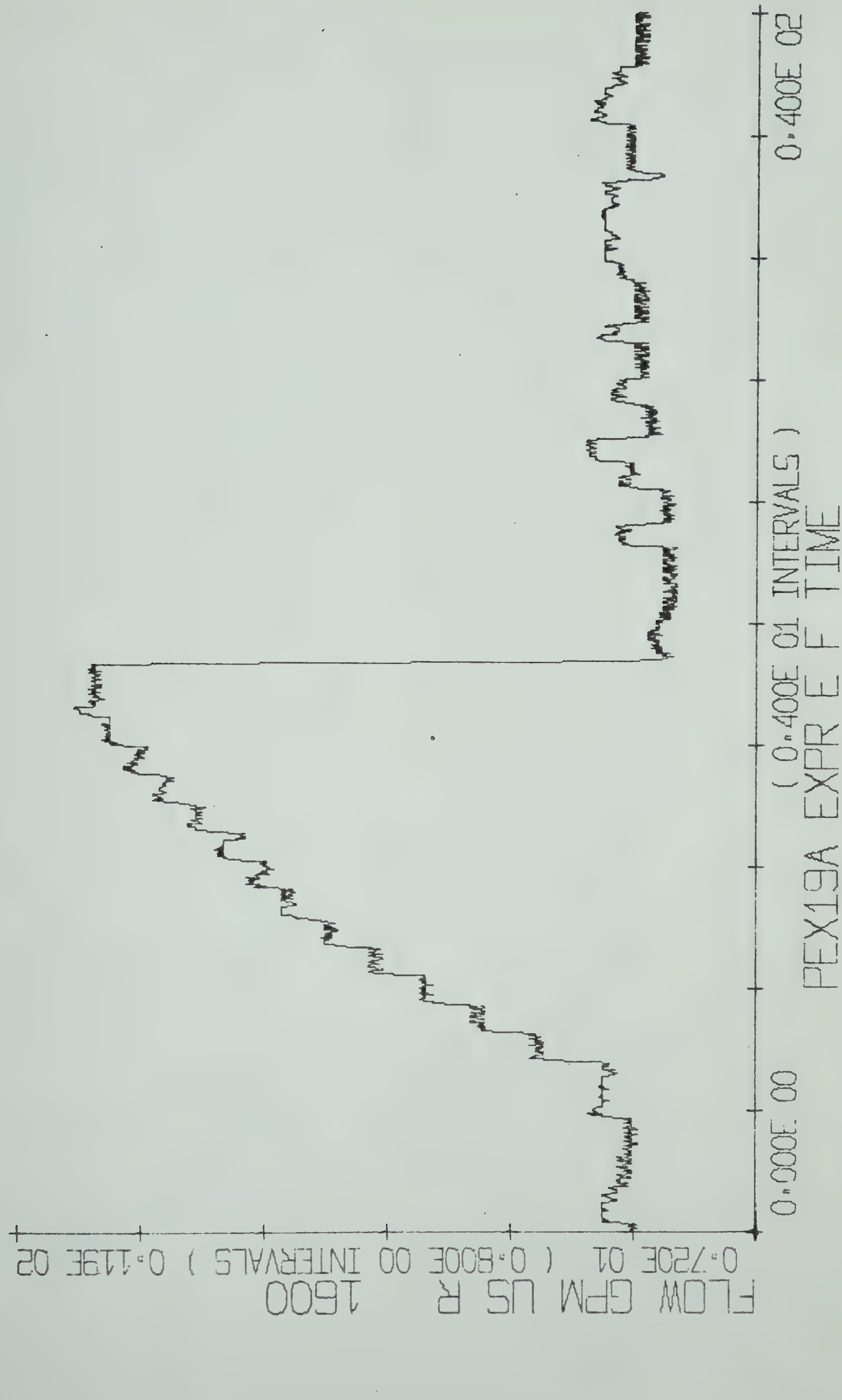
PFN13C FILN E R FREQ
(0.250E 00 INTERVALS)
0.000E 00
0.250E 01
FREQ. CONT. R 40
0.000E 00 (0.200E 00 INTERVALS) 0.100E 01

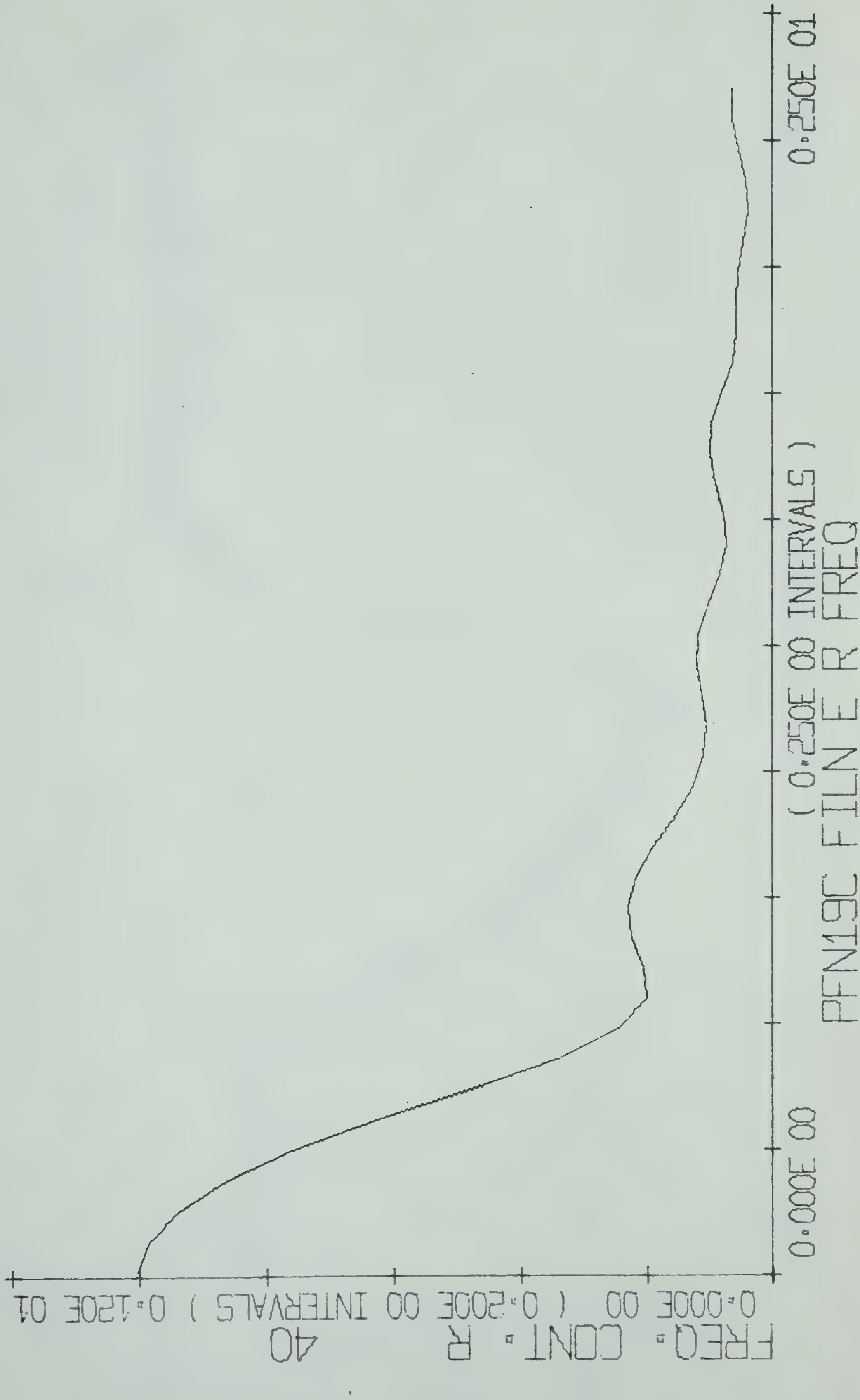

0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
FEX13B EXPR E F TIME
0.138E 03 (0.400E 00 INTERVALS) 0.138E 03
TEMP DEG F R 1600

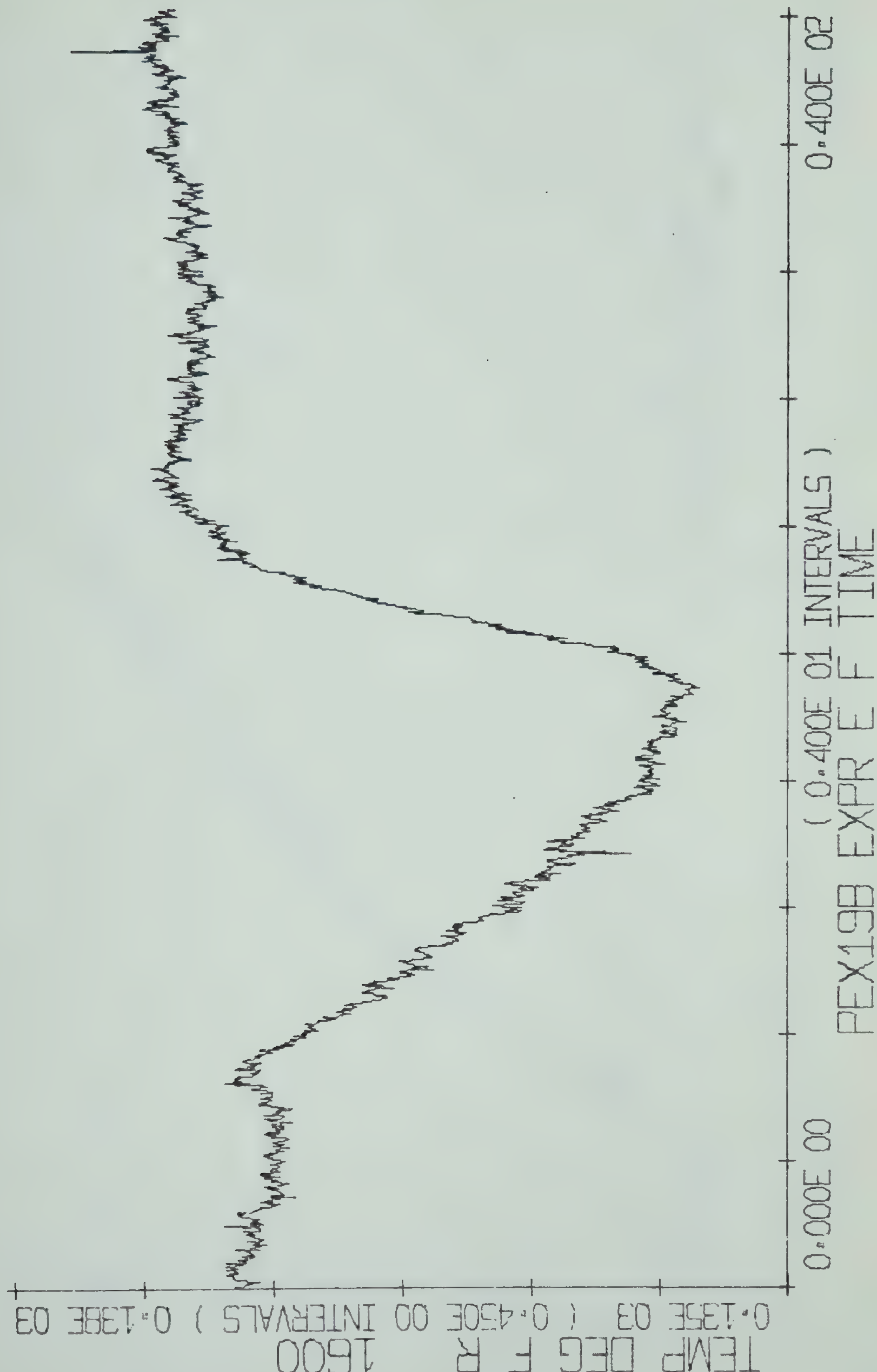


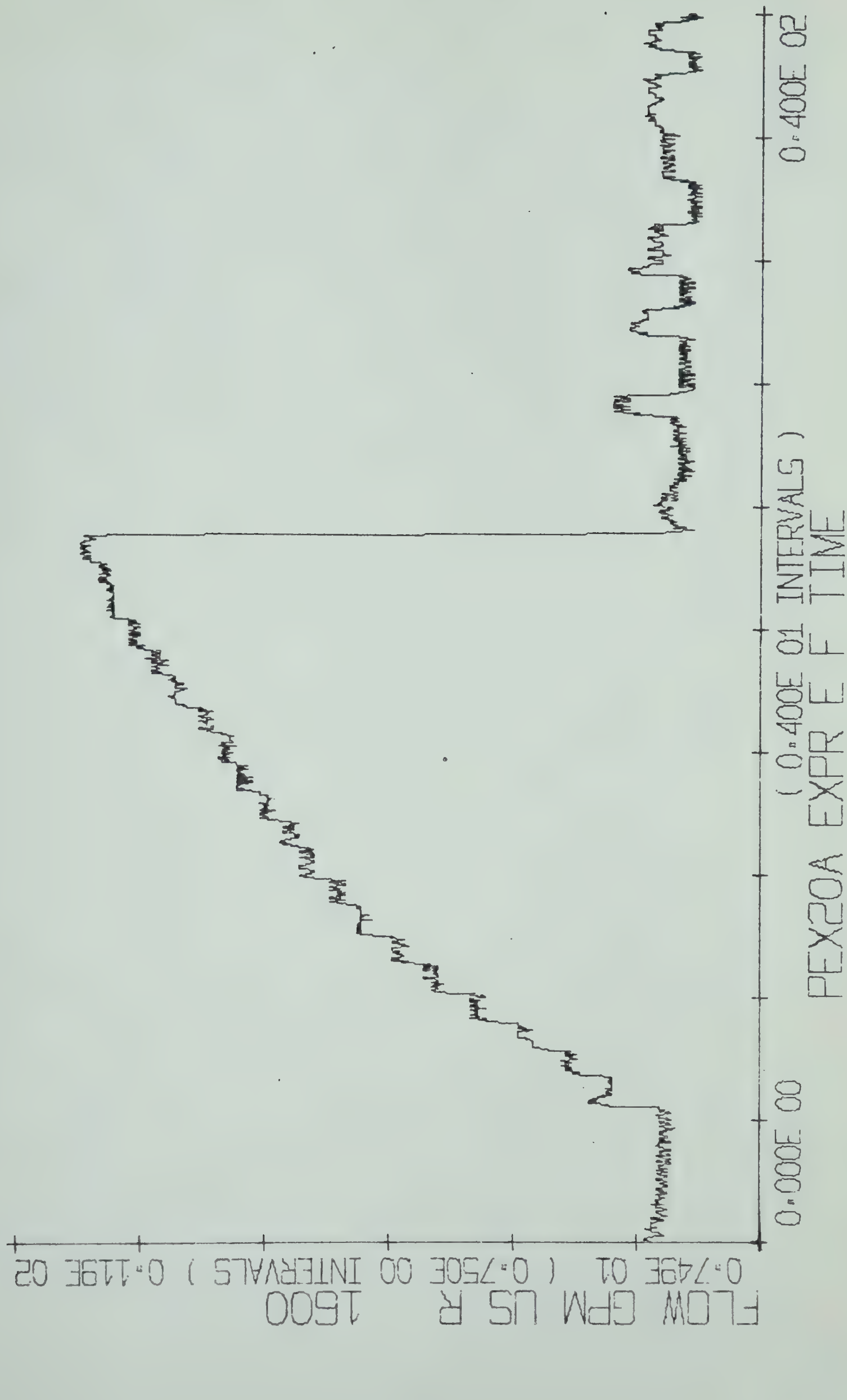
0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX13J EXPR E F TIME
0.136E 03 (0.400E 00 INTERVALS) 0.138E 03
TMP DEG F P3113 1570

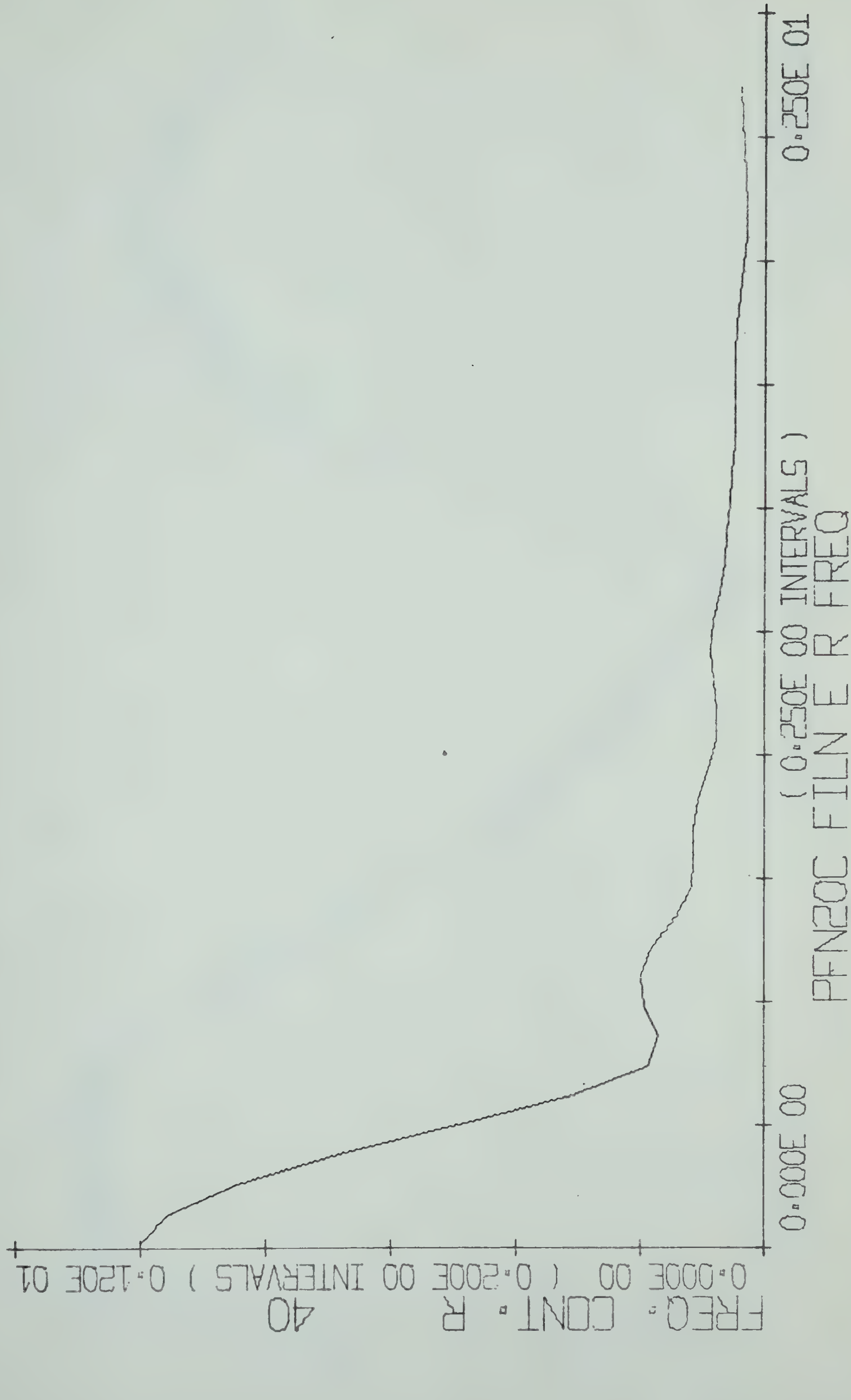

0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
PEX13D EXPR E R TIME
0.136E 03 (0.400E 00 INTERVALS) 0.138E 03
TEMP DEG F R 400

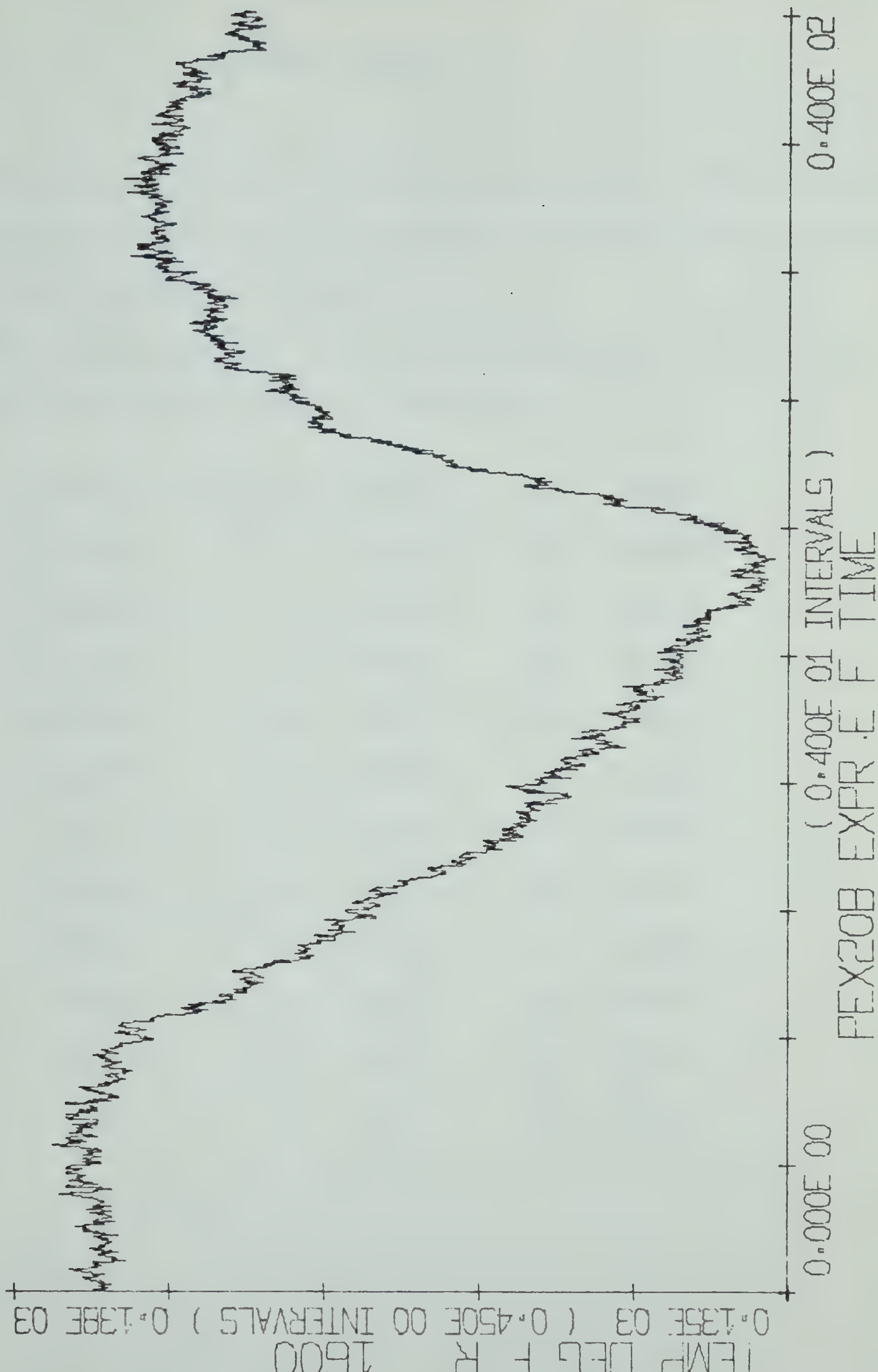

FLOW GPM US R 1600
0.720E 01 (0.800E 00 INTERVALS) 0.113E 02
0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX19A EXPR E F TIME


0.000E 00
0.250E 01
(0.250E 00 INTERVALS)
PFN19C FILN E R FREQ
0.000E 00 (0.200E 00 INTERVALS) 0.120E 01
FREQ. CONT. R
40


0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX19B EXPR E F TIME
0.135E 03 (0.450E 00 INTERVALS) 0.138E 03
TEMP DEG F R 1600


FLOW GPM US R 1500
0.749E 01 (0.750E 00 INTERVALS) 0.119E 02
0.000E 00
(0.400E 01 INTERVALS)
0.400E 02
PEX20A EXPR E F TIME


FREQ. CONT. R 40
0.000E 00 (0.200E 00 INTERVALS) 0.120E 01
0.000E 00 (0.250E 00 INTERVALS) 0.250E 01
PFN20C FILN E R FREQ



0.000E 00
0.400E 02
(0.400E 01 INTERVALS)
PEX20B EXPR E F TIME
0.135E 03 (0.450E 00 INTERVALS) 0.138E 03
TEMP DEG F R 1600

APPENDIX III

PROGRAM LISTINGS

This appendix contains commented listings of all FORTRAN computer programs used in this work. A discussion of the use of these programs can be found in Sections 4.2 and 5.4.

The programs appear in the following order where P indicates a mainline program and S indicates a subroutine:

1.	P-TSTFR	12.	P-KLTLY	23.	P-BFKTY
2.	P-WLHBL	13.	S-INITL	24.	P-BUCKWS
3.	P-PULSE	14.	S-TNFRM	25.	P-PWRSL
4.	S-TYPWR	15.	S-REVER	26.	S-NWPLT
5.	S-TMPRF	16.	P-FREQ	27.	S-NWLNS
6.	P-STDAT	17.	S-REDPT	28.	S-NWMAX
7.	P-SET	18.	S-ABCD1	29.	S-NWLGS
8.	P-CHIEF	19.	S-ABCD2	30.	S-LGRID
9.	P-VTFLO	20.	S-SUMS	31.	S-NWANN
10.	P-MVTMP	21.	S-ARPHF	32.	P-PICK
11.	P-SMOTH	22.	S-ANGLE	33.	P-WRITE
				34.	P-LINEA

PROGRAM 'TSTFR'

PROGRAM 'TSTFR'

```
C------------------------------------------------------------------
C                ** PROGRAM TSTFR **
C------------------------------------------------------------------
C
C         PURPOSE
C
C            TO GENERATE PULSE DATA WHICH IS SUITABLE FOR
C            STUDY AND EVALUATION OF FOURIER TRANSFORM
C            TECHNIQUES AS APPLIED TO PULSE TESTING.
C            THE FOLLOWING THREE TYPES OF SYSTEMS MAY BE
C            ANALYSED USING THEIR RESPONSE TO A SQUARE
C            PULSE$
C
C            1. FIRST ORDER LAG.
C            2. CASCADED FIRST ORDER LAGS.
C            3. SECOND ORDER SYSTEM.
C
C
C         PARAMETERS ENTERED VIA CARDS.
C
C            N      - NO. OF DATA POINTS GENERATED.
C            DT     - TIME INCREMENT BETWEEN DATA POINTS
C                     (SEC.).
C            IFLI   - NO. OF FILE WHERE INPUT PULSE DATA IS
C                     STORED.
C            IFLO   - NO. OF FILE WHERE OUTPUT PULSE DATA IS
C                     STORED.
C            BIAS   - = STEADY VALUE BEFORE AND AFTER PULSE.
C            RISE   - MAGNITUDE OF INPUT PULSE (SQUARE SHAPE).
C            DUR    - DURATION OF INPUT PULSE (SEC.)
C            TAU1   - TIME CONSTANT FOR -
C                     1. FIRST ORDER LAG OR
C                     2. SECOND ORDER SYSTEM.
C            TAU2   - TIME CONSTANT FOR 2ND LAG OF CASCADED
C                     LAGS.
C            IFORS  - = 1, 1ST ORDER LAG RESPONSE GENERATED.
C                     = 2, CASCADE 1ST ORDER LAG RESPONSE
C                     GENERATED.
C                     = 3, 2ND ORDER RESPONSE GENERATED.
C            ISEC   - = NO. OF RECORDS OF STEADY STATE DATA
C                     GENERATED BEFORE PULSE IS INTRODUCED.
C            NPP    - = NO. OF POINTS GENERATED AT STEADY
C                     STATE IN THE 'ISEC+1' RECORD BEFORE
C                     THE PULSE IS INTRODUCED.
C            IFLT   - NO. OF FILE WHERE TIME VALUES ARE
C                     STORED.
C            GAIN   - THE AMOUNT OF STEADY STATE GAIN ACROSS
C                     THE SYSTEM PULSED.
C
```

```
      PROGRAM 'TSTFR'
C
C          YNB    - IF 'YNB' = 'GAIN' THEN GAIN IS NOT
C                     APPLIED TO BIAS.
C                 - IF 'YNB' = 1.0 THEN GAIN IS APPLIED
C                     TO BIAS.
C          YNR    - IF 'YNR' = 'GAIN' THEN GAIN IS NOT
C                     APPLIED TO RISE.
C                 - IF 'YNR' = 1.0 THEN GAIN IS APPLIED
C                     TO RISE.
C          DR     - THE DAMPING RATIO FOR THE SECOND ORDER
C                     SYSTEM.
C
C
C        SOURCE OF EQUATIONS
C
C         BUCKLEY,P.S.,''TECHNIQUES OF PROCESS CONTROL',
C           JOHN WILEY AND SONS,INC., 1964,P.50.
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C--------------------------------------------------------------
C
C     V IS A DIMENSIONED VARIABLE IN WHICH EACH RECORD INPUT
C     , OUTPUT AND TIME VALUES ARE SAVED BEFORE STORAGE
C     ON DISK.
C
C     NPPR=NUMBER OF POINTS PER RECORD.
C
C     LUR=LOGICAL UNIT NUMBER FOR CARD INPUT.
C
C     LUW=LOGICAL UNIT NUMBER FOR LINE PRINTER OUTPUT.
C
C     DBIAS IS A DUMMY VARIABLE SET EQUAL TO THE BIAS ON
C     INPUT.
C
C     DRISE IS A DUMMY VARIABLE SET EQUAL TO THE RISE ON
C     INPUT.
C
C     NREC=NUMBER OF RECORDS OF DATA POINTS CREATED.
C
C     T=TIME IN SECONDS.
C
C     IFL=NO. OF FILE WRITTEN INTO - FIRST EQUALS 'IFLI',
C     THEN 'IFLO'.
C
C     IORO IS A FLAG INDICATING WHETHER INPUT OR OUTPUT DATA
C     IS BEING GENERATED - =1 FOR INPUT, =2 FOR OUTPUT.
C
C     IW=NO. OF RECORD IN FILE WRITE OPERATION.
C
C     IFLG IS A FLAG CHANGED FROM 1 TO 2 WHEN THE FINAL
C     STEADY STATE POINT IS GENERATED IN RECORD 'ISEC+1'.
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

PROGRAM 'TSTFR'

```
C
C
C------------------------------------------------------------------------
C
C
      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION V(40)
      GAIN=1.0
      NPPR=40
      LUR=5
      LUW=6
C     READ DATA FROM CARDS.
      READ(LUR,7)N,DT,IFLI,IFLO,BIAS,RISE,DUR,TAU1,TAU2,
     1IFORS
7     FORMAT(I5,F5.3,2I5,2F7.3,F6.2,2F10.5,I5)
      READ(LUR,500)ISEC,NPP,IFLT,GAIN,YNR,YNB,DR
500   FORMAT(3I5,4F5.2)
C     WRITE OUT DATA.
      WRITE(LUW,8) N,DT,IFLI,IFLO,BIAS,RISE,DUR,TAU1,TAU2,
     1IFORS
8     FORMAT(1X,'N=',I5,1X,'DT=',F5.3,1X,'IFLI=',I5,1X,
     1'IFLO=',I5,1X,'BIAS=',F7.3,1X,'RISE=',F7.3,1X,'DUR=',
     1F6.2,1X,'TAU1=',F10.5,1X,'TAU2=',F10.5,1X,'IFORS=',I5)
      WRITE(LUW,501)ISEC,NPP,GAIN,YNB,YNR,IFLT,DR
501   FORMAT(1X,'ISEC=',I5,1X,'NPP=',I5,1X,'GAIN=',F5.2,1X,
     1'YNB=',F5.2,1X,'YNR=',F5.2,'IFLT=',I3,'DR=',F5.3)
      DBIAS=BIAS
      DRISE=RISE
      NREC=N/NPPR+1
C     SET VARIABLES TO GENERATE INPUT DATA.
      T=0.0
      IFL=IFLI
      IORO=1
45    IW=1
C     GENERATED ISEC RECORDS OF STEADY STATE DATA.
      DO 10 K=1,ISEC
      DO 11 J=1,NPPR
      V(J)=BIAS
11    CONTINUE
```

PROGRAM 'TSTFR'

```
      WRITE(IFL'IW) V
      IW=IW+1
 10   CONTINUE
      IFLG=1
C
C     GENERATE A FURTHER NPP POINTS AT STEADY STATE, THEN
C     INITIATE THE TIME AT START OF PULSE.
C
      KSEC=ISEC+1
      DO 12 IW=KSEC,NREC
      DO 14 J=1,NPPR
      IF(IFLG-1) 20,20,21
 20   IF(J-NPP)22,22,23
 22   V(J)=BIAS
      GOTO 14
 23   IFLG=2
C
C     CHECK IF TIME IS EQUAL TO DUR.
C
 21   IF(T-DUR)25,26,26
C
C     DETERMINE WHETHER INPUT OR OUTPUT DATA IS BEING
C     GENERATED (START OF PULSE).
C
 25   GOTO(30,31),IORO
C
C     INPUT DATA
C
 30   V(J)=BIAS+RISE
      GOTO 15
C
C     OUTPUT DATA
C   **
C     DETERMINE IF SYSTEM IS 1ST ORDER LAG, CASCADED 1ST
C     ORDER LAGS OR 2ND ORDER LAG.
C
 31   GOTO(32,33,44),IFORS
C
C     FIRST ORDER.
C
 32   V(J)=RISE*(1.-EXP(-T/TAU1))+BIAS
      GOTO 15
C
C     CASCADED FIRST ORDER.
C
 33   V(J)=RISE*(1.+TAU1/(TAU2-TAU1)*EXP(-T/TAU1)-TAU2/(TAU2
     1-
     1TAU1)*EXP(-T/TAU2))+BIAS
      GOTO 15
```

```
      PROGRAM 'TSTFR'
C     SECOND ORDER.
C   **
C     INITIALIZE PARAMETERS FOR 2ND ORDER.
 44   W=1.0/TAU1*(ABS(1.0-DR**2.0))**0.5
      PHI=ATAN(((1.0-DR**2.)**0.5)/DR)
      CHI=ATAN(((1.0-DR**2.)**0.5)/DR)
C     DETERMINE IF 2ND ORDER IS UNDER, CRITICALLY OR OVER
C     DAMPED.
      IF(DR-1.0)60,61,62
C     UNDER DAMPED.
 60   V(J)=RISE*(1.0-1.0/W/TAU1*EXP(-DR*T/TAU1)*SIN(W*T+PHI)
     1)+BIAS
      GOTO 15
C     CRITICALLY DAMPED.
 61   V(J)=RISE*(1.0-(1.0+T/TAU1)*EXP(-T/TAU1))+BIAS
      GOTO 15
C     OVER DAMPED.
 62   V(J)=RISE*(1.0-DR*T/W/TAU1*EXP(-DR*T/TAU1)*
     12.*TANH((W*T+CHI)/2.0)/(1.0-TANH((W*T+CHI)/2.0)**2.0))
     2+BIAS
      GOTO 15
C     DETERMINE WHETHER INPUT OR OUTPUT DATA IS BEING
C     GENERATED (END OF PULSE).
 26   GOTO(34,35),IORO
C     INPUT DATA.
 34   V(J)=BIAS
      GOTO 15
C     OUTPUT DATA
C   **
C     DETERMINE IF SYSTEM IS 1ST ORDER LAG, CASCADED 1ST
C     ORDER LAGS OR 2ND ORDER LAG.
 35   GOTO(36,37,47),IFORS
C     FIRST ORDER SYSTEM.
```

PROGRAM 'TSTFR'

```
 36   V(J)=RISE*(1.-EXP(-T/TAU1)-1.+EXP(-(T-DUR)/TAU1))+BIAS
      GOTO 15

C     CASCADED FIRST ORDER.

 37   V(J)=RISE*(1.+TAU1/(TAU2-TAU1)*EXP(-T/TAU1)-TAU2
     1/(TAU2-TAU1)*EXP(-T/TAU2)-(1.+TAU1/(TAU2-TAU1)*
     2EXP(-(T-DUR)/TAU1)-TAU2/(TAU2-TAU1)*EXP(-(T-DUR)/TAU2)
     3))+BIAS
      GOTO 15

C     SECOND ORDER SYSTEM.
C  **
C     DETERMINE IF 2ND ORDER IS UNDER, CRITICALLY OR OVER
C     DAMPED.

 47   IF(DR-1.0)63,64,65

C     OVER DAMPED.

 63   V(J)=RISE*(1.0-1.0/W/TAU1*EXP(-DR*T/TAU1)*SIN(W*T+PHI)
     1-(1.0-1.0/W/TAU1*EXP(-DR*(T-DUR)/TAU1)*SIN(W*(T-DUR)
     2+PHI)))+BIAS
      GOTO 15

C     CRITICALLY DAMPED.

 64   V(J)=RISE*(1.0-(1.0+T/TAU1)*EXP(-T/TAU1)-(1.0-
     1(1.0+(T-DUR)/TAU1)*EXP(-(T-DUR)/TAU1)))+BIAS
      GOTO 15

C     UNDER DAMPED.

 65   V(J)=RISE*(1.0-DR*T/W/TAU1*EXP(-DR*T/TAU1)*
     12.*TANH((W*T+CHI)/2.0)/(1.0-TANH((W*T+CHI)/2.0)**2.0)-
     2(1.0-DR*(T-DUR)/W/TAU1*EXP(-DR*(T-DUR)/TAU1)**
     32.*TANH((W*(T-DUR)+CHI)/2.C)/(1.0-TANH((W*(T-DUR)+CHI)
     4/2.0)**2.0)))+BIAS
 15   T=T+DT
 14   CONTINUE

C     WRITE A RECORD OF DATA.

      WRITE(IFL'IW)V
 12   CONTINUE


      DO 999 I=1,NREC
      READ(IFL'I)V
      WRITE(LU,998)V
```

```
      PROGRAM 'TSTFR'

998   FORMAT(10(1X,F10.5))
999   CONTINUE

C     CHECK IF OUTPUT DATA HAS BEEN GENERATED.

      IF(IORO-1)40,40,41

C     SET PARAMETERS TO GENERATE OUTPUT DATA.

 40   IORO=2
      BIAS=DBIAS*GAIN/YNB
      RISE=DRISE*GAIN/YNR
      IFL=IFLO
      T=0.0
      GOTO 45

C     GENERATE FILE OF TIME VALUES.

 41   T=0.0
      DO 70 IW=1,NREC
      DO 71 J=1,NPPR
      V(J)=T
      T=T+DT
 71   CONTINUE

C     WRITE A RECORD OF TIME VALUES.

      WRITE(IFLT'IW)V
 70   CONTINUE
      CALL EXIT
      END
```

PROGRAM 'WLHBL'

```
PROGRAM 'WLHBL'

C
C-----------------------------------------------------------------
C               ** PROGRAM WLHBL **
C-----------------------------------------------------------------
C
C        PURPOSE
C
C           TO PROVIDE A HEAT BALANCE CALCULATION OF THE
C           TEST HEAT EXCHANGER WHILE ON LINE . A CORRECT
C           BALANCE INDICATES THE SYSTEM IS AT STEADY STATE.
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           ISAMPLE PERIOD = NO. OF SECONDS BETWEEN
C                            SAMPLES OF PROCESS DATA.
C
C           ITOTAL TIME = TOTAL NO. OF SECONDS OVER WHICH
C                         DATA IS SAMPLED BEFORE HEAT
C                         BALANCE IS CALCULATED.
C
C        RESTRICTIONS
C
C           1. THE PROGRAM WILL ABORT AUTOMATICALLY IF
C              (ITOTAL TIME/ISAMPLE TIME) YIELDS A
C              NUMBER GREATER THAN THE NUMBER OF STORAGE
C              LOCATIONS PERMITTED FOR THE DATA.  AT
C              PRESENT THIS IS 20.
C
C
C
C               ** GENERAL COMMENTS **
C-----------------------------------------------------------------
C
C      THIS PROGRAM IS AUTOMATICALLY CALLED BEFORE THE
C      PROGRAM 'PULSE' TO ALLOW THE OPERATOR TO ASCERTAIN
C      THAT STEADY STATE EXISTS BEFORE THE PULSE IS
C      INTRODUCED.  ITS INTERFACE WITH 'PULSE' IS EXPLAINED
C      IN THE LISTING OF 'PULSE'.
C
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C      FT=TUBE FLOW (U.S.GALS/MIN).
C
C      FS=SHELL FLOW (U.S.GALS/MIN).
C
C      TSI=INLET SHELL TEMP.(DEGREES F.).
```

```
      PROGRAM 'WLHBL'

C
C     TSO=OUTLET SHELL TEMP.(DEGREES F.).
C
C     TTI=INLET TUBE TEMP.(DEGREES F.).
C
C     TTO=OUTLET TUBE TEMP.(DEGREES F.).
C
C     LUNR=LOGICAL UNIT NUMBER FOR TYPEWRITER INPUT.
C
C     LUNW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT WHERE
C     HEAT BALANCE RESULTS APPEAR.
C
C     LUP=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT WHERE
C     DATA ENTRY INSTRUCTIONS APPEAR.
C
C     IDFT=I.D. OF TUBE FLOW LOOP.
C
C     IDFS=I.D. OF SHELL FLOW LOOP.
C
C     IDTSI=I.D. OF SHELL INLET TEMP. LOOP.
C
C     IDTSO=I.D. OF SHELL OUTLET TEMP. LOOP.
C
C     IDTTI=I.D. OF TUBE INLET TEMP. LOOP.
C
C     IDTTO=I.D. OF TUBE OUTLET TEMP. LOOP.
C
C     IGET=SAMPLING TIME (SECS).
C
C     IPRNT=TOTAL TIME (SECS).
C
C     ISAST=START TIME (TAKEN AS THE NUMBER OF SECONDS
C     ELAPSED AFTER THE LAST WHOLE HOUR).
C
C     ISCUR=CURRENT TIME IN SECONDS AFTER START.
C
C     AFT=AVG. TUBE FLOW.
C
C     AFS=AVG. SHELL FLOW.
C
C     ATTI=AVG. INLET TUBE TEMP.
C
C     ATTO=AVG. OUTLET TUBE TEMP.
C
C     ATSI=AVG. INLET SHELL TEMP.
C
C     ATSO=AVG. OUTLET SHELL TEMP.
C
C
C-----------------------------------------------------------------
```

```
      PROGRAM 'WLHBL'
C
C
      EXTERNAL PULSE
      DIMENSION FT(20),FS(20),TTI(20),TTO(20),TSI(20)
     1,TSO(20)
      LUNR=1
      LUNW=4
      LUP=2
C     DEFINE DATA ACQUISITION LOOP IDENTIFICATION NUMBERS.
      IDFT=273
      IDFS=274
      IDTSI=277
      IDTSO=279
      IDTTI=281
      IDTTO=289
C     WRITE OUT DATA ENTRY INSTRUCTIONS.
      WRITE (LUP,10)
 10   FORMAT (10X,' ENTER   ISAMPLE PERIOD,ITOTAL TIME ')
C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.
      CALL FFINP(LUNR,2,0,IGET,0,IPRNT,IEROR)
C     INITIALIZE TIME.
      I=0
      CALL TIME(JHOUR,IMIN,ISEC)
      ISAST=IMIN*60+ISEC
      ISRD=ISAST
C     CHECK WHETHER NEXT WHOLE HOUR HAS ELAPSED.
 14   CALL TIME(IHOUR,IMIN,ISEC)
      IF (IHOUR-JHOUR) 11,12,11
 11   IMIN=IMIN+60
 12   ISCUR=IMIN*60+ISEC
C     CHECK IF IT IS TIME TO SAMPLE.
      IF (ISRD+IGET-ISCUR) 13,13,14
 13   ISRD=ISCUR
      I=I+1
C     CHECK IF DIMENSION WILL BE EXCEEDED.
```

```
      PROGRAM 'WLHBL'

      IF (I-20) 15,15,50

C     CALL SUBROUTINE TO READ APPROPRIATE FLOWS AND
C     TEMPERATURES.

 15   CALL GTVLU(IDFT,1,FT(I),IERR,IF)
      CALL GTVLU(IDFS,1,FS(I),IERR,IF)
      CALL GTVLU(IDTTI,1,TTI(I),IERR,IT)
      CALL GTVLU(IDTTO,1,TTO(I),IERR,IT)
      CALL GTVLU(IDTSI,1,TSI(I),IERR,IT)
      CALL GTVLU(IDTSO,1,TSO(I),IERR,IT)

C     CHECK IF IT IS TIME TO AVERAGE VALUES.

      IF (ISAST+IPRNT-ISCUR) 16,16,14
 16   ISAST=ISCUR

C     AVERAGE THE DATA AND CALCULATE THE HEAT BALANCE

      AFT=0.0
      AFS=0.0
      ATTI=0.0
      ATTO=0.0
      ATSI=0.0
      ATSO=0.0
      DO 17 K=1,I
      AFT=AFT+FT(K)
      AFS=AFS+FS(K)
      ATTI=ATTI+TTI(K)
      ATTO=ATTO+TTO(K)
      ATSI=ATSI+TSI(K)
      ATSO=ATSO+TSO(K)
 17   CONTINUE
      AFT=AFT/I
      AFS=AFS/I
      ATTI=ATTI/I
      ATTO=ATTO/I
      ATSI=ATSI/I
      ATSO=ATSO/I

C     CALCULATE HEAT TRANSPORTS.

      QS=AFS*8.337*0.998*(ATSI-ATSO)
      QT=AFT*8.337*0.998*(ATTO-ATTI)

C     CALCULATE HEAT TRANSPORT RATIO.
C

      RATIO=QS/QT
```

```
      PROGRAM 'WLHBL'
C     WRITE OUT RESULTS
      WRITE (LUNW,18)
 18   FORMAT (/////,20X,' HEAT BALANCE RESULTS ')
      WRITE (LUNW,25)
 25   FORMAT (20X,'----------------------')
      WRITE (LUNW,19) AFT
 19   FORMAT (///,10X,' AVG.TUBE FLOW =',F5.2,'(U.S.GALS
     1/MIN)')
      WRITE (LUNW,20)AFS
 20   FORMAT (10X,' AVG.SHELL FLOW=',F5.2,'(U.S.GALS/MIN)')
      WRITE (LUNW,21) ATTI,ATTO
 21   FORMAT (//,10X,' AVG.TUBE TEMP. (DEG.F) -IN=',F5.1,
     95X,'OUT=',F5.1)
      WRITE (LUNW,22) ATSI,ATSO
 22   FORMAT (10X,' AVG.SHELL TEMP. (DEG.F)-IN=',F5.1,5X
     9'OUT=',F5.1)
      WRITE (LUNW,23) RATIO
 23   FORMAT (//,10X,' HEAT TRANSPORT RATIO (QS/QT) =',F6.3)
C     DATA SWITCH OPTION - SEE LISTING OF PROGRAM 'PULSE'.
      I=0
      CALL DATSW(7,IA)
      IF(IA-1)50,50,24
 24   CALL DATSW(8,LA)
      IF(LA-1)60,60,14
C     QUEUE PULSING PROGRAM.
 60   CALL QUEUE(PULSE,500,0)
 50   CALL VIAQ
      END
```

PROGRAM 'PULSE'

```
PROGRAM 'PULSE'

C-----------------------------------------------------------------------
C                ** PROGRAM PULSE **
C-----------------------------------------------------------------------
C
C        PURPOSE
C
C           TO INIATE AND CONTROL THE ACQUISITION OF PULSE
C           DATA FROM A HEAT EXCHANGE PROCESS.
C
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C          THE FOLLOWING PARAMETERS ARE REQUESTED AT THE
C          BEGINNING OF THE PULSE RUN WHEN PROGRAM 'WLHBL'
C          IS CALLED TO CALCULATE THE ON LINE HEAT BALANCE.
C
C           ISAMPLE PERIOD - = NO. OF SECONDS BETWEEN
C                              SAMPLES OF PROCESS DATA.
C           ITOTAL TIME    - = TOTAL NO. OF SECONDS OVER
C                              WHICH DATA IS SAMPLED BEFORE
C                              HEAT BALANCE IS CALCULATED.
C
C
C          THE FOLLOWING PARAMETERS ARE REQUESTED WHEN THE
C          PROGRAM 'PULSE' IS CALLED TO CONTROL PULSE RUN.
C
C           ISASS - NO. OF SECONDS DATA IS ACQUIRED AT
C                   STEADY STATE BEFORE FLOW IS PULSED.
C
C           ISPUL - NO. OF SECONDS OF DURATION FOR FLOW
C                   PULSE.
C
C           CHGTYP- THERE ARE FOUR POSSIBLE PULSE SHAPES--
C                  1. IF MAGNITUDE OF 'CHGTYP' IS LESS THAN
C                     MAGNITUDE OF 'CHGL1' AND 'CHGL2' THE
C                     PULSE SHAPE IS SQUARE.
C                  2. IF MAGNITUDE OF 'CHGTYP' IS GREATER
C                     THAN MAGNITUDE OF 'CHGL1' AND 'CHGL2'
C                     THE PULSE SHAPE IS A RAMP AT
C                     BEGINNING AND END.
C                  3. IF MAGNITUDE OF 'CHGTYP' IS GREATER
C                     THAN MAGNITUDE OF 'CHGL1' BUT LESS
C                     THAN 'CHGL2' THE PULSE SHAPE IS A
C                     RAMP AT BEGINNING AND A STEP AT END.
C                  4. IF MAGNITUDE OF 'CHGTYP' IS GREATER
C                     THAN 'CHGL1' BUT LESS THAN 'CHGL2'
C                     THE PULSE SHAPE IS A STEP AT
C
```

```
C                   BEGINNING AND A RAMP AT END.
C       CHGL1 - WHEN THE MAGNITUDE OF 'CHGL1' IS LESS
C               THAN THE MAGNITUDE OF 'CHTYP' ITS
C               VALUE REPRESENTS THE AMOUNT OF CHANGE
C               THAT WILL BE IMPLEMENTED AFTER EACH
C               SECOND AT THE BEGINNING OF PULSE.
C       CHGL2 - SIMILAR TO 'CHGL1' BUT APPLYING TO THE
C               END OF PULSE.
C       PLVAL - = MAXIMUM VALUE THE CHANGE WILL REACH
C               DURING THE PULSE.
C       MANUL - = 1 PULSING IS ACHIEVED BY CHANGING THE
C               SETPOINT OF THE CONTROL LOOP PULSED.
C               = 2 PULSING IS ACHIEVED BY PLACING THE
C               CONTROL LOOP PULSED ON MANUAL, AND
C               THEN CHANGING THE OUTPUT AS REQUIRED.
C
C
C      THE FOLLOWING PARAMETERS ARE REQUESTED AT THE
C      CONCLUSION OF THE PULSE RUN WHEN PROGRAM 'STDAT'
C      IS CALLED TO SORT THE ACQUIRED DATA.
C
C       IFLF  - DISK FILE NO. WHERE FLOW DATA IS TO BE
C               STORED.
C       IFLT  - DISK FILE NO. WHERE TEMP. DATA IS TO BE
C               STORED.
C       MPXF  - MULTIPLEXER NO. FOR FLOW MEASUREMENT.
C       MPXT  - MULTIPLEXER NO. FOR TEMP. MEASUREMENT.
C
C      DATA SWITCHES ARE USED AS FOLLOWS--
C
C       SWITCH 7 - ON-ABORT 'WLHBL'.
C                - OFF-LET 'WLHBL' CONTINUE.
C       SWITCH 8 - ON-ABORT 'WLHBL' CALL 'PULSE'.
C                - OFF-LET 'WLHBL' CONTINUE.
C       SWITCH 10- ON-BEFORE RUN,CONDUCT A PULSE RUN
C                  WITHOUT RAPID DATA ACQUISITION.
C                - OFF-CONDUCT PULSE RUN WITH RAPID
C                  DATA ACQUISITION.
C       SWITCH 11- ON-BEFORE RUN, MAKE OPERABLE 'RING
C                  BUFFER' LOOPS TO READ AND SAVE
C                  MEASUREMENT, SETPOINT, ERROR AND
C                  OUTPUT OF THE CONTROL LOOP PULSED.
C                - OFF-DO NOT MAKE 'RING BUFFERS'
C                  OPERABLE.
C       SWITCH 13- ON-TURNED ON DURING DATA ACQUISITION
C                  RUN WILL CAUSE IT TO ABORT.
C                - OFF-NORMAL POSITION.
C
C
C     RESTRICTIONS
```

```
      PROGRAM 'PULSE'

C
C         1. THE PROGRAM 'WLHBL' WILL AUTOMATICALLY
C            ABORT IF (ITOTAL TIME/ISAMPLE TIME) YIELDS
C            A NUMBER GREATER THAN THE NO. OF STORAGE
C            LOCATIONS PERMITTED FOR THE DATA.  AT
C            PRESENT THIS IS 20.
C
C         2. THE RAPID DATA ACQUISITION IS PRESENTLY
C            FIXED AT 40 POINTS/SEC., SAMPLED FOR A
C            PERIOD OF 40 SECONDS.
C
C         3. THE SIGN OF PARAMETERS 'CGTYP', 'CHGL1'
C            AND 'CHGL2' MUST ALWAYS BE THE SAME.
C
C         4. THE PROGRAM 'STDAT' WILL AUTOMATICALLY BE
C            CALLED IN IF DATASWITCH 10 OR 13 ARE NOT
C            TURNED ON AS SPECIFIED ABOVE.
C
C
C
C      DACS CENTRE SUBROUTINES CALLED BY PROGRAMS
C      'WLHBL', 'PULSE' AND 'STDAT' ARE--
C         FFINP, GTVLU, PTVLU, EBBIN, OPER, MMANL,
C         DATSW, TIME, VCDAQ, MAUTO, ABORT, TUCDQ,
C         FLSRT.
C
C
C             ** GENERAL COMMENTS **
C-----------------------------------------------------------
C
C     THE PROGRAM 'PULSE' WAS ORIGINALLY WRITTEN TO
C     INTRODUCE THE PULSE VIA SETPOINT CHANGES.
C     DEVELOPMENTS LED TO THE PULSING BY OUTPUT CHANGES,
C     HOWEVER THE CODING WAS NOT CHANGED SIGNIFICANTLY, SO
C     FOR THIS REASON THERE IS A DIFFERENCE BETWEEN THE
C     NAMES OF THE VARIABLES REQUESTED ON THE KEYBOARD AND
C     THE NAMES USED IN THE PROGRAM.
C
C
C         ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------
C
C     VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
C     NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C     OF THE APPROPRIATE SUBROUTINE.
C
C     IIII IS A VARIABLE REREQUIRED BY THE RAPID DATA
C     ACQUISITION PROGRAM.
C
C     ID IS A DIMENSIONED VARIABLE INTO WHICH LOOP
```

```
      PROGRAM 'PULSE'
C     IDENTIFICATION NUMBERS ARE READ IN HEXADECIMAL FORM
C     UNDER A2 FORMAT.
C
C     LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C     LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C     LOOP=NO. OF SECONDS WAITED AT THE END OF THE PULSE
C     BEFORE THE CONTROL LOOP IS RETURNED TO AUTOMATIC
C     MODE.
C
C     IDM=ID NO. FOR RING BUFFER STORING 'MEASUREMENT' OF
C     PULSED LOOP.
C
C     IDS=ID NO. FOR RING BUFFER STORING 'SETPOINT' OF
C     PULSED LOOP.
C
C     IDE=ID NO. FOR RING BUFFER STORING 'ERROR' OF
C     PULSED LOOP.
C
C     IDO=ID NO. FOR RING BUFFER STORING 'OUTPUT' OF
C     PULSED LOOP.
C
C     IFLAG IS A FLAG WHICH IS CONTINUOUSLY CHECKED BY
C     SYSTEM RAPID DATA ACQUISITION PROGRAM.  WHEN THIS
C     FLAG IS SET TO A NON ZERO VALUE RAPID DATA
C     ACQUISITION BEGINS IMMEDIATELY.
C
C     IEXIT=ERROR PARAMETER RETURNED ON CALL OF 'GTVLU' IN
C     SUBROUTINE 'TMPRF'.
C
C     IDP=ID NO. OF THE CONTROL LOOP PULSED WRITTEN IN
C     HEXADECIMAL FORM.
C
C     ID1=ID NO. OF THE CONTROL LOOP PULSED WRITTEN IN
C     BINARY FORM.
C
C     ID2=ID NO. OF LOOP READING REFERENCE BATH TEMPERATURE
C     WRITTEN IN BINARY FORM.
C
C     JOP=1 OR 2 IF DATA SWITCH 11 IS ON OR OFF.
C
C     JT=1 OR 2 IF DATA SWITCH 10 IS OR OR OFF.
C
C     JHOUR=TIME OF DAY IN HOURS AT START OF PULSE RUN.
C
C     IMIN=INITIALLY TIME OF DAY IN MINUTES AT START OF
C     PULSE RUN - CONTINUALLY UPDATED DURING RUN.
C
C     ISEC=INITIALLY TIME OF DAY IN SECONDS AT START OF
```

```
      PROGRAM 'PULSE'
C     PULSE RUN - CONTINUALLY UPDATED DURING RUN.
C
C     KSEC AT FIRST =THE TIME WHEN THE PULSE WILL BE
C     INTRODUCED IN TOTAL SECONDS AFTER THE HOUR.  LATER
C     =TIME WHEN PULSE WILL BE DELETED.
C
C     IK=2 ROUTINES 'GTVLU' AND 'PTVLU' WORK WITH SETPOINT,
C     =3 WITH OUTPUT.
C
C     SPT=VALUE OF SETPOINT OR OUTPUT (ENGINEERING UNITS)
C     OBTAINED BY 'GTVLU'.
C
C     IERR=ERROR PARAMETER ON CALL OF 'GTVLU' OR 'PTVLU'.
C     IF 'IERR' IS GREATER THAN 1 ON MORE THAN THREE
C     SUCESSIVE CALLS THEN PROGRAM EXITS.
C
C     LU=DUMMY VARIABLE REPRESENTING CODED VALUE FOR THE
C     ENGINEERING UNITS RETURNED BY GET VALUE.
C
C     IHOUR=TIME OF DAY IN HOURS AFTER THE START OF THE
C     PULSE RUN - CONTINUALLY UPDATED DURING RUN.
C
C     LSEC=TOTAL TIME AFTER THE HOUR IN SECONDS -
C     CONTINUALLY UPDATED.
C
C     NSEC=TOTAL TIME AFTER THE HOUR IN SECONDS -
C     CONTINUALLY UPDATED.  USED IN GENERATION OF RAMP
C     CHANGES EVERY SECOND.
C
C     SPCR=VALUE OF SETPOINT OR OUTPUT (ENGINEERING UNITS)
C     ENTERED WITH 'PTVLU' EACH TIME AN INCREMENT IN THE
C     RAMP IS REQUIRED.
C
C     ISEC2=TIME OF DAY IN SECONDS - CONTINUALLY UPDATED -
C     USED TO CHECK FOR LAPSE OF ONE SECOND PERIODS.
C
C     IOPER IS A PARAMETER SET TO 2 WHEN RAPID DATA
C     ACQUISITION IS COMPLETE.
C
C
C-------------------------------------------
C--------------------------
C
C

      EXTERNAL STDAT
      DIMENSION ID(2)
      COMMON IIII(376)

C     SET PARAMETERS
```

```
          PROGRAM 'PULSE'

      LUR=1
      IEXIT=1
      IDO=0107
      IDM=0108
      IDS=0109
      IDE=0110
      LOOP=3
      IFLAG=0
      IDP=0111

C     CALL SUBROUTINE TO WRITE OUT DATA ENTRY INSTRUCTIONS.

      IDK=1
      CALL TYPWR(IDK,KK,IERR,ID,AA)

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,7,0,ISASS,0,ISPUL,1,SPTI,1,SPCL1,1,
     1SPCL2,1,SPCS,0,MANUL,IEROR)

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,1,22,ID,IEROR)

C     CALL SUBROUTINE TO CONVERT HEXADECIMAL TO BINARY.

      CALL EBBIN(ID,ID1)

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,1,22,ID,IEROR)

C     CALL SUBROUTINE TO CONVERT HEXADECIMAL TO BINARY.

      CALL EBBIN(ID,ID2)

C     START RING BUFFERS WITH SWITCH 11 ON.

      CALL DATSW(11,JOP)
      IF(JOP-1)108,108,109

C     CALL SUBROUTINE TO MAKE 'OPERABLE' THE 'RING BUFFERS'
C     WHICH SAVE THE MEASUREMENT, SETPOINT, ERROR AND
C     OUTPUT OF PULSED LOOP.

 108  CALL OPER(IDM)
      CALL OPER(IDS)
```

```
          PROGRAM 'PULSE'

      CALL OPER(IDE)
      CALL OPER(IDO)

C     CALL SUBROUTINE TO READ AN AVERAGE VALUE OF THE
C     MILLIVOLTS GENERATED BY THE REFERENCE THERMOCOUPLE
C     IN THE THERMOPILE REFERENCE BATH.

 109  CALL TMPRF(ID2,RTMV1,IFLAG,IEXIT)
      IF(IEXIT-1)107,107,104

C     TEST RUN WITH DATA SWITCH 10 ON.

 107  CALL DATSW(10,JT)
      IF(JT-1)100,100,101

C     INITIALIZE RAPID DATA ACQUISITION PROGRAM.  IT WILL
C     IDLE AT THIS STAGE BECAUSE 'IFLAG=0'.

 101  CALL VCDAQ(IFLAG)

C     INITIALIZE TIME.

 100  CALL TIME(JHOUR,IMIN,ISEC)

C     SET TIME WHEN PULSE WILL BE INTRODUCED.

      KSEC=ISASS+IMIN*60+ISEC

C     CHECK IF PULSE IS TO BE INTRODUCED VIA SETPOINT
C     OR OUTPUT.

      IF(MANUL-1)201,201,202

C     VIA OUTPUT.

 202  IK=3
      GOTO 203

C     VIA SETPOINT.

 201  IK=2
 203  IDK=2
      KK=0

C     CALL SUBROUTINE TO GET THE VALUE OF THE SETPOINT
C     OR OUTPUT.

 70   CALL GTVLU(ID1,IK,SPT,IERR,LU)

C     CALL SUBROUTINE TO WRITE OUT ERROR ON GETTING, IF
```

```
      PROGRAM 'PULSE'

C     ANY.

      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)70,71,104
 71   IDK=3

C     CALL SUBROUTINE TO WRITE OUT VALUE OBTAINED.

      CALL TYPWR(IDK,KK,IERR,ID,SPT)
      IDK=4

C     CALL SUBROUTINE TO WRITE OUT VALUE PROPOSED FOR
C     PULSE.

      CALL TYPWR(IDK,KK,IERR,ID,SPCS)

C     TEST RUN WITH DATA SWITCH 10 ON.

      CALL DATSW(10,JT)
      IF(JT-1)34,34,103

C     SET FLAG TO START RAPID DATA ACQUISITION IMMEDIATELY.

 103  IFLAG=2

C     CHECK IF IT IS TIME TO PULSE FLOW.

 34   CALL TIME (IHOUR,IMIN,ISEC)
      IF(IHOUR-JHOUR)33,32,33
 33   IMIN=IMIN+60
 32   LSEC=IMIN*60+ISEC
      IF(LSEC-KSEC)34,35,35

C     PULSE FLOW.

 35   NSEC=LSEC

C     CHECK IF PULSE IS TO BE INTRODUCED VIA SETPOINT
C     OR OUTPUT.

      IF(MANUL-1)301,301,302

C     CALL SUBROUTINE TO PUT PULSED CONTROL LOOP ON MANUAL.

 302  CALL MMANL(IDP)

C     SET TIME WHEN PULSE WILL BE DELETED.

 301  KSEC=KSEC+ISPUL
      IDK=5
```

```
          PROGRAM 'PULSE'

      KK=0

C     DETERMINE IF THE BEGINNING OF PULSE IS A STEP OR RAMP.

      IF(ABS(SPTI)-ABS(SPCL1))60,60,61

C     STEP AT BEGINNING.
C   **
C     CALL SUBROUTINE TO ENTER CHANGE IN SETPOINT OR
C     OUTPUT (STEP BEGINNING).

 60   CALL PTVLU(ID1,IK,SPCS,IERR)

C     SET 'SPCR'='SPCS' IN CASE OF RAMP AT END.

      SPCR=SPCS

C     CALL SUBROUTINE TO WRITE OUT ERROR ON ENTERING,
C     IF ANY.

      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)60,62,104

C     RAMP BEGINNING.
C
C   **
C     CALCULATE FIRST VALUE TO ENTER FOR RAMP BEGINNING.

 61   SPCR=SPT+SPCL1

C     CALL SUBROUTINE TO ENTER SMALL CHANGE IN SETPOINT OR
C     OUTPUT EQUAL TO 'SPCL1' (RAMP BEGINNINGS).

 78   CALL PTVLU(ID1,IK,SPCR,IERR)

C     CALL SUBROUTINE TO WRITE OUT ERROR ON ENTERING, IF
C     ANY.

      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)78,63,104

C     CHECK IF VALUE ENTERED IS EQUAL TO MAXIMUM VALUE
C     SPECIFIED (NOTE-- PLVAL=SPCS).
C
C   **
C     CHECK IF CHANGE IS +VE OR -VE.

 63   IF(SPTI)64,64,65

C     NEGATIVE CHANGE.
```

PROGRAM 'PULSE'

```
 64   IF(SPCR-SPCS)62,62,66
C     POSTIIVE CHANGE.
 65   IF(SPCR-SPCS)66,62,62
C     CALCULATE NEXT VALUE TO ENTER FOR RAMP BEGINNING.
 66   SPCR=SPCR+SPCL1
      KK=0
C     DETERMINE IF ONE SECOND HAS ELAPSED OR IF IT IS TIME
C     TO END THE FLOW PULSE (RAMP).
 69   CALL TIME(IHOUR,IMIN,ISEC)
      IF(IHOUR-JHOUR)37,36,37
 37   IMIN=IMIN+60
 36   LSEC=IMIN*60+ISEC
C     CHECK IF PULSE DURATION HAS ELAPSED.
      IF(LSEC-KSEC)38,39,39
C     CHECK IF ONE SECOND HAS ELAPSED.
 38   IF(NSEC-LSEC)68,69,69
 68   NSEC=LSEC
      GOTO 78
C     CHECK IF IT IS TIME TO END FLOW PULSE (STEP
C     BEGINNING OR RAMP BEGINNING AFTER RAMP HAS REACHED
C     THE MAXIMUM VALUE--PLVAL)
 62   CALL TIME(IHOUR,IMIN,ISEC)
      IF(IHOUR-JHOUR)77,76,77
 77   IMIN=IMIN+60
 76   LSEC=IMIN*60+ISEC
      IF(LSEC-KSEC)62,39,39
C     DETERMINE IF THE END OF THE PULSE IS A STEP OR A RAMP.
 39   IF(ABS(SPTI)-ABS(SPCL2))80,80,81
C     STEP AT END.
C   **
C     CALL SUBROUTINE TO ENTER THE ORIGINAL VALUE OF
C     SETPOINT OR OUTPUT.
 80   CALL PTVLU(ID1,IK,SPT,IERR)
```

```
      PROGRAM 'PULSE'
C     CALL SUBROUTINE TO WRITE OUT ERROR ON ENTERING, IF
C     ANY.
      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)80,82,104
C     RAMP AT END.
C    **
C     SET THE MAXIMUM VALUE SPECIFIED EQUAL TO MAXIMUM
C     VALUE REACHED BY RAMP OR STEP AT BEGINNING.
 81   SPCS=SPCR
C     CALCULATE FIRST VALUE TO ENTER FOR RAMP AT END.
      SPCR=SPCS-SPCL2
      KK=0
C     CALL SUBROUTINE TO ENTER SMALL CHANGE IN SETPOINT OR
C     OUTPUT EQUAL TO 'SPCL2' (RAMP END).
 88   CALL PTVLU(ID1,IK,SPCR,IERR)
C     CALL SUBROUTINE TO WRITE OUT ERROR ON ENTERING,
C     IF ANY.
      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)88,20,104
C     CHECK IF VALUE ENTERED IS EQUAL TO OR HAS OVERSHOT
C     THE ORIGINAL VALUE.
C    **
C     CHECK IF CHANGE IS +VE OR -VE.
 20   IF(SPTI)84,84,85
C     POSITIVE CHANGE.
 85   IF(SPCR-SPT)86,86,87
C     NEGATIVE CHANGE.
 84   IF(SPCR-SPT)87,86,86
C     CALCULATE NEXT VALUE TO ENTER FOR RAMP AT END.
 87   SPCR=SPCR-SPCL2
      KK=0
C     CHECK IF ONE SECOND HAS ELAPSED.
```

```
      PROGRAM 'PULSE'

 94   CALL TIME (IHOUR,IMIN,ISEC2)
      IF(ISEC2-ISEC)95,94,95
 95   ISEC=ISEC2
      GOTO 88
 86   KK=0

C     CALL SUBROUTINE TO ENTER ORIGINAL VALUE OF SETPOINT
C     OR OUTPUT WHEN THE RAMP AT THE END EQUALS OR
C     OVERSHOOTS THIS VALUE,

      CALL PTVLU(ID1,IK,SPT,IERR)

C     CALL SUBROUTINE TO WRITE OUT ERROR ON ENTERING,
C     IF ANY.

      CALL TYPWR(IDK,KK,IERR,ID,AA)
      IF(KK-4)87,82,104

C     IF PULSING VIA OUTPUT CHANGE, WAIT FOR 'LOOP' SECONDS
C     BEFORE RETURNING PULSED LOOP TO AUTOMATIC CONTROL.

 82   IF(MANUL-1)204,204,205
 205  DO 220 I=1,LOOP
 219  CALL TIME(IHOUR,IMIN,ISEC2)
      IF(ISEC2-ISEC)218,219,218
 218  ISEC=ISEC2
 220  CONTINUE

C     CALL SUBROUTINE TO PLACE PULSED CONTROL LOOP ON
C     AUTOMATIC.

      CALL MAUTO(IDP)

C     DETERMINE IF RUN IS TO BE MANUALLY ABORTED.

 204  CALL DATSW(13,JA)

C     ABORT RUN WITH DATA SWITCH 13 ON.

      IF(JA-1)46,46,45

C     CALL SUBROUTINE TO ABORT RUN.

 46   CALL ABORT

C     CALL SUBROUTINE TO TEST WHETHER RAPID DATA
C     ACQUISITION HAS STOPPED AFTER 'ABORT' IS CALLED.

      CALL TVCDQ(IOPER)
      IF(IOPER-1)46,46,104
```

PROGRAM 'PULSE'

```
C      TEST RUN WITH DATA SWITCH 10 ON.
 45    CALL DATSW(10,JT)
       IF(JT-1)104,104,105
C      CALL SUBROUTINE TO TEST WHETHER RAPID DATA
C      ACQUISITION HAS COMPLETED ITS RUN.
 105   CALL TVCDQ(IOPER)
       IF(IOPER-1)82,82,51
C      CALL SUBROUTINE TO READ AN AVERAGE VALUE OF THE
C      MILLIVOLTS GENERATED BY THE REFERENCE THERMOCOUPLE
C      IN THE THERMOPILE REFERENCE BATH.
 51    CALL TMPRF(ID2,RTMV1,IFLAG,IEXIT)
       IF(IEXIT-1)106,106,104
C      CALL CHAIN TO PROGRAM 'STDAT' TO SORT ACQUIRED DATA
C      AND PLACE IT IN PERMANENT FILES.
 106   CALL CHAIN(STDAT)
 104   CALL VIAQ
       END
```

SUBROUTINE 'TYPWR'

SUBROUTINE 'TYPWR'

```
C*****************************************************************
C                 ** SUBROUTINE TYPWR **
C*****************************************************************
C
C      SUBROUTINE 'TYPWR' CONSISTS OF A COMPUTED GOTO
C      STATEMENT AND A GROUP OF WRITE STATEMENTS.
C
C
C                 ** GENERAL COMMENTS **
C*****************************************************************
C
C      THE WRITE STATEMENTS IN THIS SUBROUTINE COULD HAVE
C      BEEN INCLUDED IN THE MAINLINE OF 'PULSE', HOWEVER
C      THE SUBROUTINE WAS WRITTEN IN CASE LIMITED CORE SPACE
C      REQUIRED ALL WRITE OPERATIONS TO BE PLACED IN A
C      LOCALED SUBROUTINE.  TO ACHIVE ITS' PURPOSE ALL
C      WRITE OPERATIONS NOW IN 'PULSE' AND 'TMPRF' WOULD
C      HAVE TO BE PLACED IN THIS SUBROUTINE SO IT WOULD
C      ONLY BE CALLED BY 'PULSE'.  SINCE CORE SPACE WAS
C      SUFFICIENT THIS WAS NOT DONE.
C
C
C             ** TABLE OF FUNCTION OF VARIABLE NAMES **
C*****************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C      IDK=FLAG SET IN MAINLINE TO CALL SPECIFIC WRITE
C      STATEMENT.
C
C      KK IS A PARAMETER SET TO ZERO BEFORE EACH CALL
C      INVOLVING 'GTVLU' OR 'PTVLU'.  IF THE ERROR
C      PARAMETER 'IERR' RETURNED BY THESE ROUTINES IS OTHER
C      THAN 1, THEN KK IS INCREASED BY 1 AND THE CALL TO
C      'PTVLU' OR 'GTVLU' IS REPEATED.  IF THIS HAPPENS
C      3 TIMES KK IS SET TO 5 AND ON THE RETURN 'PULSE'
C      EXITS.  IF 'IERR' EQUALS 1 AT ANY TIME KK IS SET TO
C      4 AND 'PULSE' CONTINUES ON THE RETURN.
C
C      IERR=ERROR PARAMETER RETURNED BY SUBROUTINES 'PTVLU'
C      AND 'GTVLU'.
C
C      ID IS A DUMMY INTEGER VARIABLE WHICH CAN
C      BE TRANSFERRED FOR WRITING OUT INTEGER VALUES.
```

SUBROUTINE 'TYPWR'

```
C
C      AA IS A DUMMY REAL VARIABLE WHICH CAN
C      BE TRANSFERRED FOR WRITING OUT REAL VALUES.
C
C
C*****************************************************************
C

      SUBROUTINE TYPWR(IDK,KK,IERR,ID,AA)
      LUW=2
      GOTO(21,22,23,24,25),IDK

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

 21   WRITE(LUW,1)
 1    FORMAT('ENTER,ISASS,ISPUL,CGTYP,CHGL1,CHGL2,PLVAL,',
     1'MANUL --- PUSH EOF')
      WRITE(LUW,2)
 2    FORMAT('THEN TYPE IN THE ID OF FLOW AND REF BATH ',
     1'LOOPS RESP. --- PUSH EOF FOR EACH.')
      GOTO 50

C     CHECK THE ERROR ON 'GTVLU' AND WRITE OUT PARAMETER
C     VALUE IF NECESSARY.

 22   IF(IERR-1)5,5,6
 6    KK=KK+1
      WRITE(LUW,100)IERR,KK
 100  FORMAT('ERROR GETTING=',I3,'NO. TRY=',I3)
      IF(KK-3)50,71,71
 71   KK=5
      GOTO 50
 5    KK=4
      GOTO 50

C     WRITE OUT THE SETPOINT OR OUTPUT USED BEFORE PULSE.

 23   WRITE(LUW,101)AA
 101  FORMAT('OLD VALUE=',F10.5)
      GOTO 50
 24   WRITE(LUW,102)AA

C     WRITE OUT THE SETPOINT OR OUTPUT USED DURING PULSE.

 102  FORMAT('NEW VALUE=',F10.5)
      GOTO 50

C     CHECK THE ERROR ON 'PTVLU' AND WRITE OUT PARAMETER
C     VALUE IF NECESSARY.
```

SUBROUTINE 'TYPWR'

```
25    IF(IERR-1)7,7,8
8     KK=KK+1
      WRITE(LUW,103)IERR,KK
103   FORMAT('ERROR PUTTING=',I3,'TRY=',I3)
      IF(KK-3)50,71,71
7     KK=4
50    RETURN
      END
```

SUBROUTINE 'TMPRF'

SUBROUTINE 'TMPRF'

```
C***********************************************************
C                ** SUBROUTINE TMPRF **
C***********************************************************
C
C      SUBROUTINE 'TMPRF' READS AT 1 SECOND INTERVALS FOR 5
C      SECONDS THE MILLIVOLTAGE GENERATED BY THE REFERENCE
C      THERMOCOUPLE IN THE THERMOPILE REFERENCE BATH.  IT
C      THEN AVERAGES THESE VALUES, CALCULATES COLD JUNCTION
C      COMPENSATION AND WRITES OUT THE CHROMEL-CONSTANTAN
C      MILLIVOLTAGE EQUIVALENT TO THE TEMPERATURE IN THE
C      BATH.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C***********************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C      LRBT=ID OF LOOP READING THE RESISTANCE BULB
C      THERMOMETER ASSOCIATED WITH THE 1800 SYSTEM COLD
C      JUNCTION REFERENCE.
C
C      LREF=ID OF LOOP (DECIMAL FORM) READING THE REFERENCE
C      VOLTAGE SIGNAL ASSOCIATED WITH THE RESISTANCE BULB
C      THERMOMETER.
C
C      ID2=ID OF LOOP (DECIMAL FORM) READING THE REFERENCE
C      THERMOCOUPLE LOCATED IN THE THERMOPILE REFERENCE BATH.
C
C      IFLAG=0 BEFORE RAPID DATA ACQUISITION RUN.
C           =2 AFTER RAPID DATA ACQUISITION RUN.
C
C      IEXIT=1 - NO ERROR ON GETTING VALUES USING 'GTVLU'.
C      IEXIT=2 OR MORE ERROR OCCURRED ON GETTING VALUE.
C      WRITE OUT ERROR PARAMETER AND RETURN TO MAINLINE TO
C      EXIT FROM PROGRAM.
C
C      RTMV1=CORRECTED MILLIVOLTAGE GENERATED BY THE
C      REFERENCE THERMOCOUPLE.  THIS VALUE CORRESPONDS TO
C      THE REFERENCE BATH TEMPERATURE AS GIVEN BY
C      CHROMEL-CONSTANTAN THERMOCOUPLE TABLE.
C
C      VVOLT IS A DIMENSIONED VARIABLE IN WHICH THE
C      MILLIVOLTAGE READ FROM 'ID2' IS STORED.
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

SUBROUTINE 'TMPRF'

```
C
C     RBT IS A DIMENSIONED VARIABLE IN WHICH THE VOLTAGE
C     READ FROM 'LRBT' IS STORED.
C
C     REF IS A DIMENSIONED VARIABLE IN WHICH THE REFERENCE
C     VOLTAGE READ FROM 'LREF' IS STORED.
C
C     LK IS A COUNTER WHICH IS INCREMENTED EVERYTIME AN
C     ERROR IS ENCOUNTERED ON GETTING A VALUE WITH 'GTVLU'.
C
C     WHEN LK=3 THE SUBROUTINE RETURNS AND THE PROGRAM
C     EXITS.
C
C     IHOUR IS THE TIME OF DAY IN HOURS.
C
C     IMIN IS THE TIME OF DAY IN MINUTES.
C
C     ISEC1=TIME OF DAY IN SECONDS THE LAST TIME THE
C     SECONDS INCREMENTED.
C
C     ISEC2=PRESENT TIME OF DAY IN SECONDS.
C
C     IERR=ERROR PARAMETER ON CALL FOR 'GTVLU'.
C
C     VRBT=THE AVERAGE VALUE OF THE RESISTANCE BULB
C     THERMOMETER READING.
C
C     VREF=THE AVERAGE VALUE OF THE REFERENCE VOLTAGE.
C
C     VOLT=AVERAGE VALUE OF THE MILLIVOLTAGE READ FROM
C     THE REFERENCE THERMOCOUPLE.
C
C     TR=SYSTEM REFERENCE COLD JUNCTION BOX TEMPERATURE
C     IN DEGREES F.
C
C     VCOR=MILLIVOLTAGE CORRECTION THAT IS TO BE ADDED
C     FOR THERMOCOUPLE COLD JUNCTION COMPENSATION.
C
C
C*********************************************************
C
C

      SUBROUTINE TMPRF(ID2,RTMV1,IFLAG,IEXIT)
      DIMENSION VVOLT(5),RBT(5),REF(5)

C     INITIALIZE PARAMETERS.

      LUW=2
      LRBT=392
```

```
      SUBROUTINE 'TMPRF'

      LREF= 393
      LK=0

C     RECORD THE STARTING TIME

      CALL TIME (IHOUR,IMIN,ISEC1)
      DO 10 I=1,5

C     GET THE MEASUREMENT OF LOOPS LRBT, LREF, ID2 EVERY
C     1 SECOND FOR 5 SECONDS.

 64   CALL TIME (IHOUR,IMIN,ISEC2)

C     CHECK TIME TO SEE IF 1 SECOND HAS ELAPSED.

      IF(ISEC1-ISEC2)62,64,62
 62   ISEC1=ISEC2

C     CALL SUBROUTINE TO GET RESISTANCE BULB VOLTAGE.

 20   CALL GTVLU(LRBT,1,RBT(I),IERR,LU)

C     CHECK FOR ERROR ON GETTING RESISTANCE BULB VOLTAGE.

      IF(IERR-1)3,3,4
 4    LK=LK+1
      WRITE(LUW,100) IERR,LK
 100  FORMAT('ERROR RBT=',I3,'TRY NO=',I3)
      IF(LK-3) 20,75,75
 75   IEXIT=2
      GOTO 72
 3    LK=0

C     CALL SUBROUTINE TO GET REFERENCE VOLTAGE.

 21   CALL GTVLU(LREF,1,REF(I),IERR,LU)

C     CHECK FOR ERROR ON GETTING REFERENCE VOLTAGE.

      IF(IERR-1)5,5,6
 6    LK=LK+1
      WRITE(LUW,101) IERR,LK
 101  FORMAT('ERROR REF=',I3,'TRY NO=',I3)
      IF(LK-3) 21,75,75
 5    LK=0

C     CALL SUBROUTINE TO GET REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE.

 22   CALL GTVLU(ID2,1,VVOLT(I),IERR,LU)
```

```
      SUBROUTINE 'TMPRF'

C     CHECK FOR ERROR ON GETTING REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE.

      IF(IERR-1)7,7,8
 8    LK=LK+1
      WRITE(LUW,102) IERR,LK
 102  FORMAT('ERROR VOLT=',I3,'TRY NO=',I3)
      IF(LK-3) 22,75,75
 7    LK=0
 10   CONTINUE

C     INITIALIZE VALUES FOR AVERAGING.

      VRBT=0.0
      VREF=0.0
      VOLT=0.0

C     SUM THE READINGS OBTAINED.

      DO 90 I=1,5
      VRBT=VRBT+RBT(I)
      VREF=VREF+REF(I)
      VOLT=VOLT+VVOLT(I)
 90   CONTINUE

C     AVERAGE THE READINGS

      VRBT=VRBT/5.0
      VREF=VREF/5.0
      VOLT=VOLT/5.0

C     DIVIDE AVERAGE REFERENCE VOLTAGE BY 10 TO TAKE INTO
C     ACCOUNT THE DIFFERENCE IN EDIT CHARACTERS ON LOOPS
C     'LREF' AND 'LRBT'.

      VREF=VREF/10.0

C     WRITE OUT THE AVERAGE RESISTANCE BULB VOLTAGE.

      WRITE(LUW,6000) VRBT
 6000 FORMAT('VRBT=',F10.5)

C     WRITE OUT THE AVERAGE REFERENCE VOLTAGE.

      WRITE(LUW,6001)VREF
 6001 FORMAT('VREF=',F10.5)

C     WRITE OUT THE AVERAGE MILLIVOLTAGE ON REFERENCE
C     THERMOCOUPLE IN REFERENCE BATH.
```

```
      SUBROUTINE 'TMPRF'

      WRITE(LUW,201) VOLT
 201  FORMAT('MV=',F10.5)

C     CALCULATE TEMPERATURE IN COLD JUNCTION REFERENCE
C     BOX USING DACS SYSTEM SUPPLIED EQUATION.

      TR=51.876*VRBT/VREF+41.0
      WRITE(LUW,200) TR
 200  FORMAT('REF BOX TEMP=',F10.5)

C     CALCULATE THE COLD JUNCTION COMPENSATION MILLIVOLTAGE
C     FOR THE CHROMEL-CONSTANTAN REFERENCE THERMOCOUPLE
C     USING POLYNOMIAL DERIVED FROM A LEAST SQUARES FIT
C     TO THE TEMPERATURE-MILLIVOLTAGE TABLE FOR THE
C     RANGE 50 TO 95 DEGREES FAHRENHEIT.
C

      VCOR=((0.1507260E-04*TR)+0.3146716E-01)*TR-1.022466

C     CHECK IF THIS THE BEGINNING OR END OF DATA
C     ACQUISITION RUN.

      IF(IFLAG-1)70,70,71

C     CALCULATE CORRECTED REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE AT THE BEGINNING.

 70   RTMV1=VOLT+VCOR
      WRITE(LUW,15) RTMV1
 15   FORMAT('REF TEMP MV AT START=',F10.5)
      GOTO 72
 71   RTMV2=VOLT+VCOR

C     WRITE OUT THE CORRECTED REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE AT END OF RUN.

      WRITE(LUW,16) RTMV2
 16   FORMAT('REF TEMP MV AT END=',F10.5)

C     AVERAGE THE CORRECTED REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE FROM BEGINNING AND END OF RUN.

      RFTMV=(RTMV1+RTMV2)/2.0

C     WRITE OUT AVERAGE CORRECTED REFERENCE THERMOCOUPLE
C     MILLIVOLTAGE.

      WRITE(LUW,17) RFTMV
 17   FORMAT('AVE REF TEMP MV=',F10.5)
 72   RETURN
```

```
    SUBROUTINE 'TMPRF'
END
```

PROGRAM 'STDAT'

```
PROGRAM 'STDAT'

C-----------------------------------------------
C---------------------------
C                ** PROGRAM STDAT **
C-----------------------------------------------------------------------
C
C        PURPOSE
C
C           TO UTILIZE THE SYSTEM SUBROUTINE 'FLSRT' TO
C           SORT THE ACQUIRED DATA AND PLACE IT IN
C           PERMANENT DISK FILES.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           THE MEANING OF THE VALUES ENTERED IS EXPLAINED
C           IN THE LISTING OF PROGRAM 'PULSE'.
C
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------------
C
C     VARIABLES NOT REFERENCED IN THIS PROGRAM ARE EXPLAINED
C     IN THE SYSTEM WRITEUP FOR THE USE OF SUBROUTINE
C     'FLSRT'.
C
C     LUW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT.
C
C
C-----------------------------------------------------------------------
C
C


      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),65(41,320,U,J0)
      INTEGER OUTFL(83)
      DIMENSION IFUNC(4),INFLE(323)
      COMMON IIII(376)
      LUR=1
      LUW=2

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,111)
 111  FORMAT('ENTER,IFLF,IFLT,MPXF,MPXT')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.
```

PROGRAM 'STDAT'

```
PROGRAM 'STDAT'
      CALL FFINP(LUR,4,0,IFLF,0,IFLT,0,MPXF,0,MPXT,IEROR)
C     SORT OUTPUT DATA (TEMPERATURE).
C   **
C     SET FLAG FOR OUTPUT DATA.

      IFLAG=1
      IFUNC(1)=MPXT
      IFUNC(2)=40
      INFLE(1)=65
      INFLE(2)=320
      INFLE(3)=41
      OUTFL(1)=IFLT
      OUTFL(2)=80
      OUTFL(3)=64
      WRITE(LUW,3)
 3    FORMAT('TEMPERATURE DATA')
      CALL FLSRT(IFUNC,INFLE,OUTFL)

C     WRITE OUT ERROR PARAMETER.

 21   WRITE(LUW,1)IFUNC(3)
 1    FORMAT('ERROR ON SORT=',I3)

C     WRITE OUT LOST TIME COUNT.

      WRITE(LUW,2) IFUNC(4)
 2    FORMAT('NO. GRPS WITH TIME ERRORS=',I3//)
      IF(IFLAG-1)10,10,20

C     SORT INPUT DATA(FLOW).

 10   IFUNC(1)=MPXF
      OUTFL(1)=IFLF
      WRITE(LUW,4)
 4    FORMAT('FLOW DATA')
      CALL FLSRT(IFUNC,INFLE,OUTFL)

C     SET FLAG FOR INPUT DATA.

      IFLAG=2
      GOTO 21
 20   CALL VIAQ
      END
```

PROGRAM 'SET'

PROGRAM 'SET'

```
C-------------------------------------------------------------------------
C                 ** PROGRAM SET **
C-------------------------------------------------------------------------
C
C         PURPOSE
C
C            TO INITIALIZE THE FLAG 'IND' FOR PROGRAM 'CHIEF'
C            AND TO WRITE OUT LINKAGE SEQUENCE INSTRUCTIONS.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-------------------------------------------------------------------------
C
C     LUW=LOGICAL UNIT NO. FOR TYPEWRITTER OUTPUT.
C
C
C-------------------------------------------------------------------------
C
C

      COMMON/INSKEL/IND,IX,INPUT(15),ISGN
      IND=0
      LUW=2

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,1)
 1    FORMAT('THE VALUES TO ENTER FOR BRANCHING TO CORE',
     1'LOADS ARE AS FOLLOWS$'//)
      WRITE(LUW,8)
 8    FORMAT('HVTYP=1'/)
      WRITE(LUW,9)
 9    FORMAT('VTFLO=2'/)
      WRITE(LUW,2)
 2    FORMAT('PWRSL=3'/)
      WRITE(LUW,6)
 6    FORMAT('SMOTH=4'/)
      WRITE(LUW,3)
 3    FORMAT('KLTLY=5'/)
      WRITE(LUW,4)
 4    FORMAT('BFKTY=6'/)
      WRITE(LUW,11)
 11   FORMAT('FREQ=7'/)
      WRITE(LUW,10)
 10   FORMAT('BUCWS=8'/)
      WRITE(LUW,12)
 12   FORMAT('PICK=9'/)
      WRITE(LUW,7)
```

PROGRAM 'SET'

```
7     FORMAT('WRITE=10'/)
      WRITE(LUW,5)
5     FORMAT('SPAR1=11'/)
      WRITE(LUW,13)
13    FORMAT('ENTER A 0 AT THE END OF THE SEQUENCE')
      CALL LINK(CHIEF)
      END
```

PROGRAM 'CHIEF'

PROGRAM 'CHIEF'

```
C-----------------------------------------------------------------
C              ** PROGRAM CHIEF **
C-----------------------------------------------------------------
C
C        PURPOSE
C
C           TO PROVIDE AN EXECUTIVE TO SUPERVISE THE
C           CALLING SEQUENCE FOR THE DATA ANALYSIS PROGRAMS
C           ASSOCIATED WITH PULSE TESTING.
C
C
C
C              ** GENERAL COMMENTS **
C-----------------------------------------------------------------
C
C      THE OPERATION OF THE PROGRAM IS AS FOLLOWS$
C
C         1. THE DIMENSIONED VARIABLE INPUT IS SUPPLIED
C         WITH A SEQUENCE OF NUMBERS ENTERED FROM THE
C         KEYBOARD.
C
C         2. THE VALUE OF EACH NUMBER DETERMINES WHICH
C         PROGRAM IS CALLED.  THE ORDER OF THE NUMBERS
C         DETERMINES THE ORDER IN WHICH THE PROGRAMS ARE
C         CALLED.
C
C         3. WHEN A ZERO IS ENCOUNTERED IN 'INPUT' THE
C         PROGRAM REQUESTS A NEW SEQUENCE OR EXITS DEPENDING
C         UPON THE VALUE OF 'ISGN'.
C
C         4. ALL ANALYSIS PROGRAMS RETURN TO CHIEF ON
C         COMPLETION.
C
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C      IND IS A FLAG INITIALLY SET TO ZERO BY THE
C      INITIALIZATION PROGRAM 'SET'.
C
C      IX IS THE SUBSCRIPT FOR THE LOCATION OF THE SEQUENCE
C      VALUE BEING CONSIDERED.
C
C      ISGN=1 A NEW SEQUENCE IS REQUESTED WHEN A ZERO IS
C      ENCOUNTERED IN 'INPUT', =2 PROGRAM EXITS WHEN ZERO IS
C      ENCOUNTERED.
C
C      JUMP IS SET EQUAL TO EACH NUMBER IN THE SEQUENCE
```

PROGRAM 'CHIEF'

```
C     OF 'INPUT' SO IT CAN BE USED AS THE INDEX IN A
C     COMPUTED GO TO FOR SELECTION OF THE NECESSARY
C     PROGRAM.
C
C
C-----------------------------------------------------------------
C
C

      COMMON/INSKEL/IND,IX,INPUT(15),ISGN
      LUR=1
      LUW=2
      IF(IND)10,10,1000
 10   IX=1

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,5)
 5    FORMAT('ENTER LINKAGE SEQUENCE FROM INSTRUCTIONS ABOVE
     1')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,1,150,INPUT,IEROR)

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,8)
 8    FORMAT('ENTER 1 FOR RETURN, 2 FOR EXIT')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,1,0,ISGN,IEROR)
      IND=1

C     CHECK FOR A ZERO IN SEQUENCE.

      IF(INPUT(IX))100,100,11
 1000 IX=IX+1

C     CHECK FOR A ZERO IN SEQUENCE.

      IF(INPUT(IX))100,100,11

C     SET GO TO INDEX.

 11   JUMP=INPUT(IX)
```

```
      PROGRAM 'CHIEF'
C     SELECT THE REQUESTED PROGRAM.
      GOTO (21,22,23,24,25,26,27,28,29,30,31),JUMP
 21   CALL LINK(MVTMP)
 22   CALL LINK(VTFLO)
 23   CALL LINK(PWRSL)
 24   CALL LINK(SMOTH)
 25   CALL LINK(KLTLY)
 26   CALL LINK(BFKTY)
 27   CALL LINK(FREQ)
 28   CALL LINK(BUCWS)
 29   CALL LINK(PICK)
 30   CALL LINK(WRITE)
 31   CALL LINK(SPAR1)
C     CHECK IF A FURTHER SEQUENCE IS TO BE ENTERED.
 100  IF(ISGN-1)10,10,500
 500  CALL EXIT
      END
```

PROGRAM 'VTFLO'

PROGRAM 'VTFLO'

```
C
C-----------------------------------------------------------------
C                ** PROGRAM VTFLO **
C-----------------------------------------------------------------
C
C       PURPOSE
C          1. TO CONVERT THE INTEGER VALUES READ FROM THE
C             FLOW TURBINE TO THE EQUIVALENT IN VOLTS BY
C             READING THEM A RECORD AT A TIME FROM DISK
C             FILES.
C
C          2. TO CONVERT THE VOLTAGE VALUES OBTAINED IN 1
C             TO THEIR EQUIVALENT IN GALLONS/MINUTE (U.S.)
C             AND STORE THE RESULTS IN A NEW DISK FILE.
C
C
C       PARAMETERS ENTERED VIA KEYBOARD
C
C          IFLR  - NO. OF FILE WHERE SORTING PROGRAM
C                  'STDAT' HAS STORED 'FLOW DATA'
C                  (INTEGER FORM).
C          IFLW  - NO. OF FILE WHERE FLOW DATA IN GALLONS
C                  PER MINUTE (U.S.) IS TO BE STORED
C                  (REAL FORM).
C          N     - NUMBER OF POINTS THAT ARE TO BE CONVERT-
C                  ED TO FLOW.
C          IFLOC - TURBINE IDENTIFICATION, =1 - 1 INCH SIZE
C                  =2 - 3/4 INCH SIZE.
C
C       RESTRICTION
C
C          1. THE FILE NUMBERS IFLR AND IFLW MUST BE
C             DIFFERENT.
C
C
C          ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C
C     IVOLT IS A DIMENSIONED VARIABLE INTO WHICH
C     INTEGER VALUES REPRESENTING THE VOLTAGE FROM THE FLOW
C     TURBINE ARE READ A RECORD AT A TIME.
C
C     FLOW IS A DIMENSIONED VARIABLE IN WHICH THE
C     CONVERTED FLOW VALUES ARE SAVED.
C
C     LUR=LOGICAL UNIT NUMBER FOR TYPEWRITER INPUT
C
```

```
      PROGRAM 'VTFLO'

C     LUW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT.
C
C     NPPRI=NO. OF INTEGER POINTS PER RECORD WHICH IS
C     EQUAL TO NO. OF WORDS PER RECORD.
C
C     VOLTR=VOLTAGE RANGE ON THE MULTIPLEXER POINT
C     MEASURING THE FLOW TURBINE SIGNAL.
C
C     NPPRF=NO. OF REAL POINTS PER RECORD.
C
C     IRL=NO. OF LAST RECORD OF INTEGER VALUES USED.
C
C     ISW AND IFW REPRESENT TWO CONSECUTIVE RECORD NUMBERS -
C     THE RECORDS WHERE THE REAL VALUES SAVED IN 'FLOW' ARE
C     STORED.
C
C     J IS A SUBSCRIPT FOR THE LOCATION OF THE FIRST DATA
C     POINT TO BE STORED IN A RECORD.
C
C     K IS SUBSCRIPT FOR THE LOCATION OF THE LAST DATA
C     POINT TO BE STORED IN A RECORD.
C
C
C-------------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION IVOLT(80),FLOW(160)

C     INITIALIZE PARAMETERS.

      LUR=1
      LUW=2
      NPPRI=80
      VOLTR=5.0
      NPPRF=NPPRI/2
      WRITE(LUW,5555)
 5555 FORMAT('THIS IS PROGRAM VTFLO'/)

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,1)
 1    FORMAT('ENTER,IFLR,IFLW,N,IFLOC')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.
```

```
      PROGRAM 'VTFLO'

      CALL FFINP(LUR,4,0,IFLR,0,IFLW,0,N,0,IFLOC,IEROR)
      IRL=N/NPPRI
      ISW=0
      IFW=0
      DO 10 IR=1,IRL
C     READ A RECORD OF INTEGER VALUES REPRESENTING VOLTAGE.
      READ(IFLR'IR)IVOLT
      DO 11 I=1,NPPRI
C     CHANGE THE INTEGER VALUE TO A REAL VALUE.
      V=FLOAT(IVOLT(I))
C     CALCULATE THE ACTUAL VOLTAGE VALUE MEASURED.
      V=VOLT1/32767.0*V
C     CALCULATE THE FLOW METER FREQUENCY BY USING A LEAST
C     SQUARES POLYNOMIAL FIT TO A FREQUENCY-VOLTAGE TABLE.
C     THEN APPLY A LINEAR CONVERSION CONSTANT TO CONVERT
C     FREQUENCY INTO GALLONS PER MINUTE (U.S.).
      IF(IFLOC-1)15,15,16
 15   FLOW(I)=(((0.870309*V-13.249)*V+288.714)*V+9.52604)
     1*0.0426
      GOTO 11
 16   FLOW(I)=(((0.916809*V-13.2838)*V+287.047)*V+11.0247)
     1*0.0195
 11   CONTINUE
      ISW=IFW+1
      IFW=ISW+1
C     INTEGER VALUES YIELDS TWO RECORDS OF REAL FLOW
C     VALUES.
      J=1
      K=NPPRF
      DO 10 IW=ISW,IFW
      WRITE(IFLW'IW) (FLOW(L),L=J,K)
      J=J+NPPRF
      K=2*K
 10   CONTINUE
      CALL LINK(CHIEF)
      END
```

PROGRAM 'MVTMP'

PROGRAM 'MVTMP'

```
C-----------------------------------------------------------------------
C              ** PROGRAM MVTMP **
C-----------------------------------------------------------------------
C
C        PURPOSE
C
C           1. TO CONVERT THE INTEGER VALUES READ FROM THE
C           THERMOPILE TO THE EQUIVALENT IN MILLIVOLTS BY
C           READING THEM FROM A DISK FILE A RECORD AT A
C           TIME.
C
C           2. TO CONVERT THE MILLIVOLTAGE VALUES OBTAINED
C           IN 1. TO THEIR EQUIVALENT IN DEGREES FAHRENHEIT,
C           AND TO STORE THE RESULTS IN A NEW FILE.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           IFLR  - NO. OF FILE WHERE SORTING PROGRAM
C                   'STDAT' HAS STORED 'TEMPERATURE DATA'
C                   (INTEGER FORM).
C           IFLW  - NO. OF FILE WHERE TEMPERATURE DATA IN
C                   DEGREES FAHRENHEIT IS TO BE STORED
C                   (REAL FORM).
C           N     - NO. OF POINTS TO BE CONVERTED TO
C                   TEMPERATURE.
C           PROBC - NO. ELEMENTS IN THE THERMOPILE USED.
C           RFTMV - AVERAGE REFERENCE BATH REFERENCE
C                   THERMOCOUPLE MILLIVOLTAGE AS WRITTEN
C                   OUT BY PROGRAM 'PULSE'.
C                   (COMPENSATED MILLIVOLTAGE FOR
C                   THERMOCOUPLE MEASURING REFERENCE BATH
C                   TEMPERATURE).
C
C
C        RESTRICTIONS
C
C           1. THE FILE NOS. IFLR AND IFLW MUST BE
C              DIFFERENT.
C
C
C              ** GENERAL COMMENTS **
C-----------------------------------------------------------------------
C
C      THE ORIGINAL THERMOPILE WAS MADE OF CHROMEL-
C      CONSTANTAN (CH-C) THERMOCOUPLE WIRE.  THE TEMPERATURE
C      OF THE THERMOPILE REFERENCE BATH WAS MEASURED BY A
C      SINGLE REFERENCE THERMOCOUPLE OF THE SAME MATERIAL.
```

```
      PROGRAM 'MVTMP'

C     TO CONVERT THE SIGNAL MEASURED FROM THE THERMOPILE TO
C     DEGREES F INVOLVED THE FOLLOWING STEPS --
C
C        1. CONVERT INTEGER VALUE READ FROM THERMOPILE
C        MULTIPLEXER TO ITS EQUIVALENT IN MILLIVOLTS
C           USING A LINEAR CONVERSION.
C        2. DIVIDE THE MILLIVOLT VALUE BY THE NUMBER OF
C           PROBES AND ADD IT TO 'RFTMV' - THE COMPENSATED
C           MILLIVOLTAGE FROM THE REFERENCE THERMOCOUPLE.
C        3. SUBSTITUTE THE RESULT OF 2. INTO A POLYNOMIAL
C           DERIVED BY A LEAST SQUARES FIT TO THE CH-C
C           THERMOCOUPLE TABLES TO OBTAIN THE TEMPERATURE
C           AT THE THERMOPILE HOT JUNCTIONS.
C
C     FAILURE OF THE CH-C THERMOPILE FORCED THE USE OF A
C     COPPER-CONSTANTAN (C-C) TYPE.  BECAUSE OF THIS
C     TWO ADDITIONAL LEAST SQUARES POLYNOMIAL FITS TO THE
C     C-C TABLES WERE REQUIRED.  BASICALLY THE 'RFTMV'
C     MILLIVOLTAGE VALUE MEASURED WITH A CH-C TYPE
C     HAD TO BE CONVERTED TO THE C-C MILLIVOLTAGE
C     EQUIVALENT FOR THE SAME TEMPERATURE.  THE STEPS
C     INVOLVED CONVERTING THE CH-C MILLIVOLTAGE TO DEGREES
C     F THEN THE DEGREES F TO THE EQUIVALENT C-C
C     MILLIVOLTAGE.
C
C
C          ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C     MVOLT IS A DIMENSIONED VARIABLE INTO WHICH INTEGER
C     VALUES REPRESENTING THE MILLIVOLTS READ FROM THE
C     THERMOPILE ARE READ A RECORD AT A TIME.
C
C     TEMP IS A DIMENSIONED VARIABLE IN WHICH THE VALUES
C     CONVERTED TO TEMPERATURE ARE SAVED.
C
C     LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C     LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C     NPPRI=NO. OF INTEGER POINTS PER RECORD WHICH IS EQUAL
C     TO NO. WORDS PER RECORD.
C
C     IRL=NO. OF LAST RECORD OF INTEGER VALUES USED.
C
C     ISW AND IFW REPRESENT TWO CONSECUTIVE RECORD NUMBERS -
C     THE RECORDS WHERE THE REAL VALUES SAVED IN 'TEMP'
C     ARE STORED.
C
C     NPPRF=NO. OF REAL POINTS PER RECORD.
```

```
      PROGRAM 'MVTMP'
C
C     J IS A SUBSCRIPT FOR THE LOCATION OF THE FIRST DATA
C     POINT TO BE STORED IN A RECORD.
C
C     K IS SUBSCRIPT FOR THE LOCATION OF THE LAST DATA
C     POINT TO BE STORED IN A RECORD.
C
C     X IS THE TOTAL MILLIVOLTAGE WHICH IS REPRESENTATIVE
C     OF THE TEMPERATURE AT THE HOT JUNCTION OF THE
C     THERMOPILE.
C
C     Z IS A DUMMY VARIABLE EQUAL TO 'RFTMV' USED FOR
C     SUBSTITUTION IN A POLYNOMIAL.
C
C     T EQUALS THE TEMPERATURE IN THE THERMOPILE REFERENCE
C     BATH IN DEGREES F.
C
C     Y IS THE MILLIVOLTAGE THAT WOULD BE GENERATED BY A
C     COPPER-CONSTANTAN THERMOCOUPLE IF THE HOT
C     JUNCTION WAS AT TEMPERATURE 'T' AND THE COLD
C     JUNCTION WAS AT ICE POINT.
C
C
C-----------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION MVOLT(80),TEMP(80)

C     INITIALIZE PARAMETERS.

      LUR=1
      LUW=2
      NPPRI=80
      WRITE(LUW,5555)

C   **
C     WRITE OUT PROGRAM IDENTIFICATION.

 5555 FORMAT('THIS IS PROGRAM MVTMP'/)

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,1)
 1    FORMAT('ENTER,IFLR,IFLW,N,PROBC,RFTMV')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
```

```
      PROGRAM 'MVTMP'

C     ENTRY.

      CALL FFINP(LUR,5,0,IFLR,0,IFLW,0,N,1,PROBC,1,RFTMV
     1,IEROR)
      IRL=N/NPPRI
      ISW=0
      IFW=0
      NPPRF=NPPRI/2
      Z=RFTMV

C     CONVERT MILLIVOLTAGE VALUE 'RFTMV' TO THE EQUIVALENT
C     TEMPERATURE IN DEGREES F USING A LEAST SQUARES
C     POLYNOMIAL FIT TO THE CHROMEL-CONSTANTAN THERMOCOUPLE
C     TABLE.

      IF(Z-4.26298)88,88,98
 88   T=(((0.8351718E-02*Z)-0.4344516)*Z+30.833308)*Z+
     132.001089
      GOTO 13
 98   T=(((0.2150481E-02*Z)-0.2575562)*Z+29.656109)*Z+
     134.299718
 13   IF(T-91.0)33,33,34

C     CONVERT THE DEGREES F EQUIVALENT TO MILLIVOLTS USING
C     LEAST SQUARES POLYNOMIAL FIT TO THE COPPER-
C     CONSTANTAN THERMOCOUPLE TABLE.

 33   Y=(((0.1482909E-07*T)+0.1081611E-04)*T+0.2067943E-01)
     1*T-0.[illegible]
      GOTO 19
 34   IF(T-151.0)43,43,44
 43   Y=(((0.7703282E-08*T)+0.9890805E-05)*T+0.2104733E-01)
     1*T-0.[illegible]
      GOTO 19
 44   Y=(((0.1301623E-07*T)+0.4450828E-05)*T+0.2235491E-01)
     1*T-0.[illegible]

C     WRITE OUT TEMPERATURE IN REFERENCE BATH.

 19   WRITE(LUW,100) T
 100  FORMAT('TEMP REF BATH=',F10.5)
      DO 10 IR=1,IRL

C     READ A RECORD OF THE INTEGER VALUES REPRESENTING
C     MILLIVOLTS.

      READ(IFLR'IR) MVOLT
      DO 11 I=1,NPPRI

C     CHANGE INTEGER VALUE TO A REAL VALUE.
```

```
          PROGRAM 'MVTMP'

      V=FLOAT(MVOLT(I))

C     CALCULATE THE ACTUAL MILLIVOLTAGE VALUE MEASURED.
C
C     MEASUREMENT IS MADE ON A 20 MV RANGE.
C
C     ADD THE MILLIVOLTAGE FROM THE THERMOPILE TO 'Y' TO
C     OBTAIN MILLIVOLTAGE EQUIVALENT TO THE TEMPERATURE AT
C     THE THERMOPILE HOT JUNCTIONS.

      X=Y+20.0/32767.0*V/PROBC

C     OBTAIN TEMPERATURE IN DEGREES F AT THE THERMOPILE
C     HOT JUNCTION BY USING A LEAST SQUARES POLYNOMIAL FIT
C     TO THE COPPER-CONSTANTAN THERMOCOUPLE TABLE.

      IF(X-1.309)20,20,21
 20   TEMP(I)=(((0.6025596E-02*X)-1.252428)*X+46.69154)*X
     1+32.01969
      GOTO 11
 21   IF(X-2.736)22,22,23
 22   TEMP(I)=(((0.1702397E-01*X)-1.046060)*X+46.05431)*X+
     132.48098
      GOTO 11
 23   TEMP(I)=(((-0.6593423E-02*X)-0.6461391)*X+44.35243)*X
     1+34.63365
 11   CONTINUE
      ISW=IFW+1
      IFW=ISW+1
      J=1
      K=NPPRF
      DO 10 IW=ISW,IFW

C     STORE THE TEMPERATURE VALUES ON DISK.  NOTE ONE
C     RECORD OF INTEGER VALUES YIELDS TWO RECORDS OF REAL
C     TEMPERATURE VALUES.

      WRITE(IFLW'IW)  (TEMP(L),L=J,K)
      J=J+NPPRF
      K=2*K
 10   CONTINUE
      CALL LINK(CHIEF)
      END
```

PROGRAM 'SMOTH'

PROGRAM 'SMOTH'

```
C-----------------------------------------------------------------------
C              ** PROGRAM SMOTH **
C-----------------------------------------------------------------------
C
C        PURPOSE
C           TO PROVIDE A VARIETY OF DIGITAL SMOOTHING
C           FORMULAS FOR DATA ANALYSIS OPERATIONS.
C           THE FOLLOWING FORMULAE ARE AVAILABLE --
C
C         1 SMOTHING WITH LEAST SQUARES FIT TO POLYNOMIALS
C           OF ORDERS 2 TO 5 INCLUSIVE,USING FROM 5 TO 13
C           POINTS INCLUSIVE.
C
C         2 EXPONENTIAL SMOOTHING (SYMMETRICAL) USING 3,5 OR
C           7 POINTS INCLUSIVE.
C
C         3 SMOOTHING OF SPECTRAL DENSITIES USING ONE OF
C           THREE SPECTRAL WINDOW FORMULAS.
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           IFLR  - NO. OF DISK FILE CONTAINING RAW DATA.
C           IFLW  - NO. OF DISK FILE WHERE SMOOTHED DATA
C                   IS TO BE STORED.
C           N     - TOTAL NO. OF POINTS TO BE SMOOTHED.
C           IFLTP - SELECTS THE TYPE OF FILTERING USED,
C                   =1 - LEAST SQUARES, =2 - EXPONENTIAL,
C                   =3 - SPECTRAL WINDOW.
C           IDEG  - SELECTS THE ORDER OF POLYNOMIAL USED IN
C                   LEAST SQUARES FIT, =1 - 2ND OR 3RD,
C                   =2 - 4TH OR 5TH.
C           NS    - DETERMINES THE NO. OF POINTS USED --
C                   IFLTP=1,IDEG=1,NS=1,2,3 OR 4 FOR 5,7,9
C                   OR 11 POINTS.
C                   IFLTP=1,IDEG=2,NS=1,2,3 OR 4 FOR 7,9,11
C                   OR 13 POINTS.
C                   IFLTP=2,NS=1,2 OR 3 FOR 3,5 OR 7 POINTS.
C           B     - EXPONENTIAL SMOOTHING CONSTANT.
C                   AS B GOES FROM 1.0 TO 0.0 THE WEIGHT
C                   GIVEN TO THE POINTS EITHER SIDE OF
C                   POINT BEING SMOOTHED GOES FROM 1.0 TO
C                   0.0 .
C           ISW   - SELECTS SPECTRAL WINDOW FORMULA,
C                   =1 - HANNING, =2 - HAMMING, =3 - HANNING
C                   SECOND PASS.
C           NLP   - = NUMBER OF TIMES THE PROGRAM WILL
C                   EXECUTE USING THE SAME PARAMETERS.
C                   THIS CAN BE USED WHEN SMOOTHING IS
```

```
      PROGRAM 'SMOTH'

C                 APPLIED MORE THAN ONCE TO THE SAME
C                 DATA SET.
C          NFINC - = NO. OF FILE NO. INCREMENTS.  THIS MAY
C                 BE USED WHEN CONSECUTIVELY NUMBERED
C                 FILES ARE TO BE SMOOTHED USING THE
C                 SAME PARAMETERS.  NOTE BOTH NOS. OF
C                 IFLR AND IFLW ARE INCREASED BY 1 WITH
C                 EACH EXECUTION OF 'SMOTH' UNTIL 'NFINC'
C                 INCREMENTS HAVE BEEN MADE.
C          LFLAG - =1, RETURN TO SMOOTH ANOTHER FILE,
C                 =2, EXIT FROM PROGRAM.
C
C
C      RESTRICTIONS
C
C          THE VALUES SMOOTHED MUST BE REAL NUMBERS.
C
C              ** GENERAL COMMENTS **
C-------------------------------------------------------------
C
C     THREE OPTIONS FOR SMOOTHING TECHNIQUES MAY BE
C     SELECTED USING 'IFLTP'.  WHEN PARAMETERS REQUESTED
C     HAVE NO SIGNIFICANCE FOR THE PARTICULAR TECHNIQUE
C     SELECTED, A 1 SHOULD BE ENTERED FOR THEIR VALUE.
C     THEREFORE IF 'IFLTP'=3, 'IDEG', 'NS' AND 'B' HAVE
C     NO MEANING.
C
C     IFLR AND IFLW MAY BE DIFFERENT OR THE SAME DEPENDING
C     UPON WHETHER OR NOT THE UNSMOOTHED DATA IS TO BE SAVED
C     OR DESTROYED.
C
C
C          ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-------------------------------------------------------------
C
C     DATAR IS A DIMENSIONED VARIABLE LARGE ENOUGH TO
C     HOLD TWO RECORDS OF DATA.  THE RAW DATA VALUES ARE
C     READ FROM DISK INTO THIS DIMENSION.
C
C     DATAS IS A DIMENSIONED VARIABLE LARGE ENOUGH TO
C     HOLD ONE RECORD OF DATA.  THE SMOOTHED DATA VALUES
C     ARE SAVED IN THE DIMENSION OF THIS VARIABLE.
C
C     PN IS A DIMENSIONED VARIABLE IN WHICH THE VALUES
C     INVOLVED IN SMOOTHING CALCULATIONS ARE SAVED.
C     THE DIMENSION OF PN IS SUFFICIENT TO ACCOMODATE THE
C     LARGEST NUMBER OF POINTS INVOLVED IN ANY SMOOTHING
C     CALCULATION.
C
C     LUR=LOGICAL UNIT NUMBER FOR TYPEWRITER INPUT.
```

PROGRAM 'SMOTH'

```
C
C     LUW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT.
C
C     NPPR=NUMBER OF POINTS PER RECORD.
C
C     NPLST=NO. OF POINTS LOADED INTO PN VECTOR AT START.
C
C     JFIL IS A DUMMY VARIABLE IN THE LOOP THAT INCREMENTS
C     THE FILE NUMBERS.
C
C     JLP IS A DUMMY VARIABLE IN THE LOOP THAT APPLIES THE
C     SAME SMOOTHING CALCULATIONS TO THE DATA MORE THAN
C     ONCE.
C
C     IRL=NO. OF LAST RECORD OF RAW DATA USED.
C
C     NN=NO. PTS. USED IN SMOOTHING CALCULATION.
C
C     I IS SUBSCRIPT FOR EACH SMOOTHED VALUE AS IT IS
C     CALCULATED.
C
C     IR REPRESENTS THE NO. OF THE RECORD OF SMOOTHED
C     DATA CURRENTLY BEING DETERMINED.
C
C     SUMN IS AN INTERMEDIATE VALUE IN THE SMOOTHING
C     CALCULATIONS.
C
C     IRR=NO. OF RECORD OF RAW DATA READ.
C
C
C----------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION DATAR(40),DATAS(40),PN(15)

C     INITIALIZE PARAMETERS.

      LUR=1
      LUW=2
      NPPR=40

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,4)
 4    FORMAT('THIS IS THE SMOOTHING ROUTINE')
```

```
      PROGRAM 'SMOTH'

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

 1000 WRITE(LUW,1)
 1    FORMAT('ENTER,IFLR,IFLW,N,IFLTP,IDEG,NS,B,ISW,NLP,
     1NFINC,LFLAG')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,11,0,IFLR,0,IFLW,0,N,0,IFLTP,0,IDEG,
     10,NS,1,B,0,ISW,0,NLP,0,NFINC,0,LFLAG,IEROR)

C     SET THE NUMBER OF POINTS TO BE LOADED INTO THE PN
C     VECTOR AT THE START.

      GOTO(301,301,303),IFLTP
 301  GOTO(5,7,9,11),NS
 5    NPLST=4
      GOTO 2
 7    NPLST=6
      GOTO 2
 9    NPLST=8
      GOTO 2
 11   NPLST=10
 2    IF(IFLTP-1)3,3,45
 3    IF(IDEG-1)12,12,32
 12   NPLST=NPLST+2
      GOTO 32
 45   NPLST=NPLST-2
      GOTO 32
 303  NPLST=2
      NN=NPLST+1

C     START OF LOOP WHICH INCREMENTS FILE NUMBER.

 32   DO 500 JFIL=1,NFINC

C     START OF LOOP CONTROLLING THE NUMBER OF TIMES
C     SMOOTHING IS APPLIED.

      DO 50 JLP=1,NLP
      IRL=N/NPPR+1

C     READ THEFIRST RECORD OF RAW DATA.

      READ(IFLR'1)DATAR

C     INITIALIZATION (FORM PN VECTOR).

      DO 13 I=2,NN
```

PROGRAM 'SMOTH'

```
      J=I-1
      PN(I)=DATAR(J)
 13   CONTINUE
      DO 20 IR=1,IRL
C
C     START SMOOTHING CALCULATIONS
C     SMOOTH POINT BY POINT FOR ONE RECORD.
C
      DO 21 I=1,NPPR
C
C     CHECK IF THIS IS THE FIRST RECORD OF THE VALUES
C     BEING SMOOTHED.
C
      IF(IR-1) 39,39,69
C
C     CHECK FOR THE LAST RAW POINT TO BE PLACED IN FIRST
C     RECORD OF SMOOTHED DATA.
C
 39   IF(I-NN/2)250,250,69
C
C     SET THE FIRST 'NPLST' SMOOTHED POINTS EQUAL TO THE RAW
C     POINTS.
C
 250  DATAS(I)=DATAR(I)
      GOTO 21
 69   DO 30 K=1,NPLST
C
C     SHIFT DATA IN PN VECTOR ONE SPACE BACK.
C
      KA=K+1
      PN(K)=PN(KA)
 30   CONTINUE
C
C     ADD A NEW RAW DATA VALUE IN THE LAST LOCATION OF NP.
C
      PN(NN)=DATAR(I)
C
C     SMOOTHING FORMULA
C   **
C     SELECT EXPONENTIAL, LEAST SQUARES OR SPECTRAL WINDOW.
C
      GOTO (44,46,47),IFLTP
C
C     LEAST SQUARES.
C   **
C     SELECT THE ORDER OF THE POLYNOMIAL FITTED.
C     QUDRATIC OR CUBIC.
C
 44   GOTO(101,102),IDEG
```

PROGRAM 'SMOTH'

```
C       SELECT THE NO. OF PTS. USED IN SMOOTHING CALCULATION.

 101    GOTO(105,107,[illegible],110),NS

C       EXPONENTIAL.
C    **
C       SELECT THE NO. OF PTS. USED IN SMOOTHING CALCULATION.

 46     GOTO (131,132,133),NS

C       SELECT SPECTRAL WINDOW FORMULA.

 47     GOTO(304,305,306),ISW

C       HANNING SPECTRAL WINDOW.

 304    DATAS(I)=0.5*PN(2)+0.25*(PN(1)+PN(3))
        GOTO 55

C       HAMMING SPECTRAL WINDOW.

 305    DATAS(I)=0.54*PN(2)+0.23*(PN(1)+PN(3))
        GOTO 55

C       HANNING SECOND PASS.

 306    DATAS(I)=0.[illegible](2)+0.1[illegible](PN(1)+PN(3))
        GOTO 55

C       THREE POINT EXPONENTIAL.

 131    SUMN=B*PN(2)+(B**2)*(PN(1)+PN(3))
        DATAS(I)=SUMN/(B+2*B**2)
        GOTO 55

C       FIVE POINT EXPONENTIAL.

 132    SUMN=B*PN(3)+(B**2)*(PN(2)+PN(4))+(B**3)*(PN(1)+PN(5))
        DATAS(I)=SUMN/(B+2*(B**2+B**3))
        GOTO 55

C       SEVEN POINT EXPONENTIAL.

 133    SUMN=B*PN(4)+(B**2)*(PN(3)+PN(5))+(B**3)*
       1(PN(2)+PN(6))+(B**4)*(PN(1)+PN(7))
        DATAS(I)=SUMN/(B+2*(B**2+B**3+B**4))
        GOTO 55

C       SECOND AND THIRD ORDER WITH FIVE POINTS
```

```
      PROGRAM 'SMOTH'

 105  SUMN=17.*PN(3)+12.*(PN(2)+PN(4))-3.*(PN(1)+PN(5))
      DATAS(I)=SUMN/35.0
      GOTO 55

C     SECOND AND THIRD ORDER WITH SEVEN POINTS.

 107  SUMN=7.*PN(4)+6.*(PN(3)+PN(5))+3.*(PN(2)+PN(6))
     1-2.*(PN(1)+PN(7))
      DATAS(I)=SUMN/21.0
      GOTO 55

C     SECOND AND THIRD ORDER WITH NINE POINTS.

 109  SUMN=59.*PN(5)+54.*(PN(4)+PN(6))+39.*(PN(3)+PN(7))+
     114.*(PN(2)+PN(8))-21.*(PN(1)+PN(9))
      DATAS(I)=SUMN/231.0
      GOTO 55

C     SECOND AND THIRD ORDER WITH ELEVEN POINTS.

 111  SUMN=89.*PN(6)+84.*(PN(5)+PN(7))+69.*(PN(4)+PN(8))+44.
     1*(PN(3)+PN(9))+9.*(PN(2)+PN(10))-36.*(PN(1)+PN(11))
      DATAS(I)=SUMN/429.0
      GOTO 55

C     QUARTIC AND QUINTIC .
C   **
C     SELECT THE NO. OF PTS. USED IN SMOOTHING CALCULATION.

 102  GOTO(207,209,211,213),NS

C     FOURTH AND FIFTH ORDER WITH SEVEN POINTS.

 207  SUMN=131.*PN(4)+75.*(PN(3)+PN(5))-30.*(PN(2)+PN(6))
     1+5.*(PN(1)+PN(7))
      DATAS(I)=SUMN/231.0
      GOTO 55

C     FOURTH AND FIFTH ORDER WITH NINE POINTS.

 209  SUMN=179.*PN(5)+135.*(PN(4)+PN(6))+30.*(PN(3)+PN(7))
     1-55.*(PN(2)+PN(8))+15.*(PN(1)+PN(9))
      DATAS(I)=SUMN/429.0
      GOTO 55

C     FOURTH AND FIFTH ORDER WITH ELEVEN POINTS.

 211  SUMN=143.*PN(6)+120.*(PN(5)+PN(7))+60.*(PN(4)+PN(8))
     1-10.*(PN(3)+PN(9))-45.*(PN(2)+PN(10))+18.*(PN(1)+
     1PN(11))
```

PROGRAM 'SMOTH'

```
      DATAS(I)=SUMN/429.0
      GOTO 55
C     FOURTH AND FIFTH ORDER WITH THIRTEEN POINTS.
213   [illegible]=679.*PN(7)+600.*(PN(6)+PN(8))+390.*(PN(5)+[illegible](9))+
     1110.*(PN(4)+PN(10))-160.*(PN(3)+PN(11))-198.*
     1(PN(2)+PN(12))+110.*(PN(1)+PN(13))
      DATAS(I)=[illegible]/2431.0
55    CONTINUE
[illegible] CONTINUE
      WRITE(IFLW'IR)DATAS
      IRR=IR+1
      READ(IFLR'IRR)DATAR
[illegible] CONTINUE
[illegible] CONTINUE
C     INCREMENT FILE NUMBERS FOR MULTIPLE FILE SMOOTHING IF
C     [illegible].
      IFLR=IFLR+1
      IFLW=IFLW+1
[illegible] CONTINUE
      GOTO(14,15),LFLAG
C     RESET LAST RECORD INDICATOR.
14    GOTO 1000
15    CALL LINK (CHIEF)
      END
```

PROGRAM 'KLTLY'

```
PROGRAM 'KLTLY'

C-----------------------------------------------------------------------
C                   ** PROGRAM KLTLY **
C-----------------------------------------------------------------------
C
C         PURPOSE
C
C             TO CALCULATE THE FOURIER TRANSFORM OF TIME
C             DOMAIN INPUT AND OUTPUT PULSE DATA USING THE
C             COOLEY-TUKEY FAST FOURIER TRANSFORM TECHNIQUE.
C
C
C         PARAMETERS ENTERED VIA DEYBOARD
C
C             IFLX  - FILE NO. FOR INPUT PULSE DATA (TIME).
C             IFLY  - FILE NO. FOR OUTPUT PULSE DATA (TIME).
C             IFLRI - FILE NO. FOR REAL PART OF TRANSFORM
C                     OF INPUT PULSE (FREQ.).
C             IFLII - FILE NO. FOR IMAG. PART OF TRANSFORM OF
C                     INPUT PULSE (FREQ.).
C             IFLRO - FILE NO. FOR REAL PART OF TRANSFORM OF
C                     OUTPUT PULSE (FREQ.).
C             IFLIO - FILE NO. FOR IMAG. PART OF TRANSFORM OF
C                     OUTPUT PULSE (FREQ.).
C             N     - NO. OF TIME DOMAIN POINTS USED IN
C                     TRANSFORM (MUST=A POWER OF 2).
C             NM    - =POWER OF 2 WHICH YIELDS N (2**NM=N).
C             RATE  - =THE PULSE DATA SAMPLE RATE
C                     (POINTS/SECOND).
C             IRASS - =NO. OF RECORDS OF TIME DATA WHICH
C                     WILL BE USED TO DETERMINE THE STEADY
C                     STATE BEFORE PULSE OCCURED.
C
C
C         RESTRICTIONS
C
C             1. ALL FILE NUMBERS SPECIFIED ABOVE SHOULD BE
C                DIFFERENT.
C             2. N MUST EQUAL A POWER OF 2.
C             3. THE MINIMUM FREQUENCY VALUE WHICH MAY BE
C                DETERMINED IS A FUNCTION OF 'N' AND 'DT'.
C                A LOWER MINIMUM IS ACHIEVED BY INCREASING
C                THE MAGNITUDE OF EITHER OF THESE VARIABLES.
C
C
C             ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------------
C
C      VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
```

```
      PROGRAM 'KLTLY'
C
C     NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C     OF THE APPROPRIATE SUBROUTINE.
C
C     THE FOLLOWING 6 VARIABLES APPLY TO THE COOLEY-TUKEY
C     TRANSFORM 'TNFRM'--
C
C
C     IX=THE STORAGE LOCATION ADDRESS AS ESTABLISHED IN
C     SUBROUTINE 'TNFRM'.
C
C     IXR=THE BIT REVERSED STORAGE LOCATION ADDRESS AS
C     DETERMINED BY THE BIT REVERSAL ASSEMBLY LANGUAGE
C     SUBROUTINE 'REVER'.
C
C     AR=REAL NUMBER INPUT - OUTPUT ARRAY (DIMENSION N).
C
C     AI=IMAG. NUMBER INPUT - OUTPUT ARRAY (DIMENSION N).
C
C     INV=1, TRANSFORM CALCULATED, =2, INVERSE TRANSFORM
C     CALCULATED.
C
C     INP=1 REAL INPUT DATA ONLY, =2 COMPLEX INPUT DATA.
C
C
C     LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C     LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C     NPPR=NO. OF POINTS PER RECORD IN DISK FILES.
C
C     IFLAG=1 FOR ANALYSIS OF OUTPUT DATA, =2, FOR ANALYSIS
C     OF INPUT DATA.
C
C     NRECW=NO. OF RECORDS OF FOURIER TRANSFORM RESULTS.
C
C     KL IS SUBSCRIPT FOR LAST LOCATION FROM WHICH A RECORD
C     OF DATA IS CHOOSEN FOR STORAGE ON DISK.
C
C     KF IS SUBSCRIPT FOR FIRST LOCATION FROM WHICH A
C     RECORD OF DATA IS CHOSEN FOR STORAGE ON DISK.
C
C
C-----------------------------------------------------------
C
C
      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      COMMON N,NM,IX,IXR,AR(1040),AI(1040),INV,INP
```

```
      PROGRAM 'KLTLY'

      COMMON LUR,LUW,IFLX,IFLY,IFLRI,IFLII,IFLRO,IFLIO,RATE

C     INITIALIZE PARAMETERS.

      LUR=1
      LUW=2
      INV=1
      INP=1
      NPPR=40

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,5555)
 5555 FORMAT('THIS IS FOURIER TRANSFORM ROUTINE---KLTLY')

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,10)
 10   FORMAT('ENTER,IFLX,IFLY,IFLRI,IFLII,IFLRO,IFLIO,N,',
     1'NM,RATE,IRASS')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,10,0,IFLX,0,IFLY,0,IFLRI,0,IFLII,0,
     1IFLRO,0,IFLIO,0,N,0,NM,1,RATE,0,IRASS,IEROR)

C     SET UP ANALYSIS OF OUTPUT DATA.

      IFLAG=1

C     CALL SUBROUTINE TO INITIALIZE PARAMETERS AND DATA.

      CALL INITL(IFLAG,IFL2,IFL3,NREC,IRASS)

C     CALL COOLEY-TUKEY SUBROUTINE TO OBTAIN REAL AND
C     IMAGINARY PARTS OF FOURIER TRANSFORM.

 15   CALL TNFRM

C     WRITE ON DISK THE REAL AND IMAGINARY COMPONENTS OF THE
C     COMPLEX NUMBERS OF THE TRANSFORM.

      NRECW=NREC-IRASS
      KL=0
      DO 13 IRW=1,NRECW
      KF=KL+1
      KL=KL+NPPR
      WRITE(IFL2'IRW)(AR(LL),LL=KF,KL)
      WRITE(IFL3'IRW) (AI(LL),LL=KF,KL)
```

```
      PROGRAM 'KLTLY'

 13   CONTINUE

C     DETERMINE WHETHER INPUT DATA HAS BEEN ANALYSED.

      IF(IFLAG-1) 22,22,33
 22   IFLAG=2

C     SET UP ANALYSIS OF INPUT DATA.

      CALL INITL(IFLAG,IFL2,IFL3,NREC,IRASS)
      GOTO 15
 33   CALL LINK (CHIEF)
      END
```

SUBROUTINE 'INITL'

```
SUBROUTINE 'INITL'

C*****************************************************************
C                 ** SUBROUTINE INITL **
C*****************************************************************
C
C      SUBROUTINE 'INITL' CALCULATES THE STEADY STATE,
C      SUBTRACTS THE STEADY STATE FROM THE PULSE DATA,
C      CALCULATES THE AREA UNDER THE INPUT PULSE, AND
C      DETERMINES THE ERROR OF CLOSURE FOR BOTH THE INPUT
C      AND OUTPUT PULSES.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C****************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      SUM=VALUE OF THE STEADY STATE.
C
C      NREC=TOTAL NO. OF RECORDS OF TIME DATA USED IN
C      DETERMINING STEADY STATE AND FOURIER TRANSFORM.
C
C      DT=INCREMENT BETWEEN TIME DATA POINTS (SEC.).
C
C      IFL1=NO. OF FILE CONTAINING PULSE DATA.
C
C      IFL2=NO. OF FILE CONTAINING REAL PART OF TRANSFORM.
C
C      IFL3=NO. OF FILE CONTAINING IMAG. PART OF TRANSFORM.
C
C      KL IS SUBSCRIPT FOR LAST LOCATION INTO WHICH A RECORD
C      OF TIME DATA IS READ.
C
C      KF IS SUBSCRIPT FOR FIRST LOCATION INTO WHICH A
C      RECORD OF TIME DATA IS READ.
C
C      NAVE=NO. OF TIME DATA POINTS USED TO DETERMINE STEADY
C      STATE.
C
C      IRTNF=NO. OF FIRST RECORD OF TIME DATA USED IN
C      FOURIER TRANSFORM.
C
C      NECL IS SUBSCRIPT OF THE LOCATION WHERE ERROR OF
C      CLOSURE CALCULATIONS BEGIN.
C
C      NN IS SUBSCRIPT FOR THE LOCATION OF THE 2ND LAST TIME
C      POINT USED IN TRANSFORM.
```

SUBROUTINE 'INITL'

```
C
C     N4 IS SUBSCRIPT FOR THE LOCATION OF THE 3RD LAST
C     TIME POINT USED IN TRANSFORM.
C
C     N2 IS SUBSCRIPT FOR THE LOCATION OF THE 4TH LAST TIME
C     POINT USED IN TRANSFORM.
C
C
C*************************************************************
C
C

      SUBROUTINE INITL(IFLAG,IFL2,IFL3,NREC,IRASS)
      COMMON N,NM,IX,IXR,AR(1040),AI(1040),INV,INP
      COMMON LUR,LUW,IFLX,IFLY,IFLRI,IFLII,IFLRO,IFLIO,RATE
      NPPR=40
      SUM=0.0

C     CALCULATE NO. OF RECORDS OF TIME DOMAIN DATA USED
C     CONSIDERING THE DATA USED TO CALCULATE STEADY STATE
C     WILL NOT BE USED IN FOURIER TRANSFORM.

      NREC=N/NPPR+1+IRASS
      DT=1.0/RATE

C     DETERMINE IF INPUT OR OUTPUT DATA IS BEING ANALYSED.

      IF(IFLAG-1) 111,111,112

C     SET FILE NOS. FOR OUTPUT DATA.

 111  IFL1=IFLY
      IFL2=IFLRO
      IFL3=IFLIO
      GOTO 17

C     SET FILE NOS. FOR INPUT DATA.

 112  IFL1=IFLX
      IFL2=IFLRI
      IFL3=IFLII
 17   KL=0

C     READ TIME DOMAIN DATA FROM DISK TO CALCULATE STEADY
C     STATE.

      DO 9 IRR=1,IRASS
      KF=KL+1
      KL=KF+NPPR-1
      READ(IFL1'IRR) (AR(LL),LL=KF,KL)
```

SUBROUTINE 'INITL'

```
 9    CONTINUE
C     BEGIN CALCULATION OF STEADY STATE.
      NAVE=IRASS*NPPR
      DO  8 I=1,NAVE
C     SUM 'NAVE' STEADY STATE VALUES.
      SUM=SUM+AR(I)
 8    CONTINUE
C     AVERAGE SUM TO OBTAIN STEADY STATE.
      SUM=SUM/NAVE
C     DETERMINE IF STEADY STATE APPLIES TO OUTPUT OR INPUT.
      IF(IFLAG-1)19,19,20
C     WRITE OUT VALUE OF STEADY STATE FOR OUTPUT.
 19   WRITE(LUW,21) SUM
 21   FORMAT('STEADY STATE OUTPUT=',F10.5)
      GOTO 22
C     WRITE OUT VALUE OF STEADY STATE FOR INPUT.
 20   WRITE(LUW,23) SUM
 23   FORMAT('STEADY STATE INPUT=',F10.5)
C     READ THE TIME DOMAIN DATA TO BE USED IN FOURIER
C     TRANSFORM.
 22   KL=0
      IRTNF=IRASS+1
      DO 11 IRR=IRTNF,NREC
      KF=KL+1
      KL=KF+NPPR-1
      READ(IFL1'IRR)(AR(LL),LL=KF,KL)
 11   CONTINUE
C     SUBTRACT THE STEADY STATE FROM ALL THE VALUES TO
C     OBTAIN A CLOSED PULSE.
      DO 12 I=1,N
      AR(I)=AR(I)-SUM
 12   CONTINUE
C     WRITE OUT ERROR OF CLOSURE EVERY 3RD POINT IN THE
```

```
      SUBROUTINE 'INITL'

C     LAST RECORD OF TIME DOMAIN DATA.
C   **
C     DETERMINE IF INPUT OR OUTPUT DATA IS BEING ANALYSED.

      IF(IFLAG-1)27,27,28
 27   WRITE(LUW,24)
 24   FORMAT('ERROR OF CLOSURE ON OUTPUT DATA')
      GOTO30
 28   WRITE(LUW,31)
 31   FORMAT('ERROR OF CLOSURE ON INPUT DATA')
 30   NECL=N-NPPR+1

C     WRITE OUT ERROR OF CLOSURE.
C   **
C     SINCE STEADY STATE WAS SUBTRACTED ABOVE, THE VALUES OF
C     'AR' AT THE END OF THE DATA SET REPRESENT THE ERROR OF
C     CLOSURE.

      DO 25 I=NECL,N,3
      WRITE(LUW,26) AR(I)
 26   FORMAT(1X,F10.5)
 25   CONTINUE
      IF (IFLAG-1) 106,106,105

C     CALCULATE THE AREA UNDER THE INPUT CURVE USING
C     SIMPSON'S RULE.
C
C     SINCE 'N' IS A POWER OF 2 IT WILL ALWAYS BE AN EVEN
C     NUMBER, THEREFORE THERE WILL ALWAYS BE AN ODD NUMBER
C     OF SUBINTERVALS.  TO OBTAIN THE EVEN NO. NECESSARY
C     FOR SIMPSON'S RULE, THE LAST DATA POINT IS NEGLECTED.
C     FOR A CLOSED PULSE THIS VALUE WILL BE ZERO SO LITTLE
C     ERROR WILL OCCUR IN THE ERROR CALCULATION.

 105  NN=N-1

C     SET AREA EQUAL TO SUM OF FIRST AND LAST POINT
C     CONSIDERED.

      AREA=AR(1)+AR(NN)
      N4=NN-1

C     MULTIPLY EVERY 2ND POINT BY 4 STARTING WITH THE 2ND
C     POINT UP TO 2ND LAST POINT CONSIDERED.

      DO 100 I=2,N4,2
      AREA=AREA+AR(I)*4.0
 100  CONTINUE
      N2=NN-2
```

```
      SUBROUTINE 'INITL'

C     MULTIPLY EVERY 2ND POINT BY 2 STARTING WITH THE
C     3RD POINT UP TO 3RD LAST POINT CONSIDERED.

      DO 101 I=3,N2,2
      AREA=AREA+AR(I)*2.0
 101  CONTINUE

C     CALCULATE THE TOTAL AREA.

      AREA=AREA*DT/3.0

C   **
C     WRITE OUT THE AREA UNDER THE INPUT CURVE.

      WRITE(LUW,102) AREA
 102  FORMAT('AREA UNDER INPUT CURVE=',F10.5)
 106  RETURN
      END
```

COLLEY-TUKEY TRANSFORM

AS WRITTEN BY

TRINITY UNIVERSITY

COMPUTER CENTER

The FORTRAN Subroutine TNFRM uses two two-point transforms plus a N/4 point transform to decrease computation time of a discrete Fourier transform. To conserve space the intermediate and final values calculated use the input area, thus, maximizing use of storage.

Parameters of the subroutine are:

N - the number of data points which must be a power of 2.

NM - the exponent of 2, i.e., $N=2^{NM}$

AR(I) - the real input and output array, which has dimensions of N

AI(I) - which may be imaginary input but is always theimaginary output array

INV - if = 1, it indicates real input data only; if = 2, it indicates complex input only

These parameters must be established in the main program before the subroutine is called. Two other parameters which appear in common are IX and IXR which are the storage location address and the bit reversed storage location address. IXR is determined by the bit reversal assembly language subroutine REVER.

The power density is computed by taking the transform of the data and multiplying the transform by its complex conjugate and dividing by 1/DT, where DT is the time between samples. The user must put in his own print and format statements

The A array will contain the frequencies which correspond to the transformed points 1 to N in correct order. The user should note that points of the transform beyond F (i) = 1/2DT will not be valid because of the sampling theorem, i.e., the input data cannot contain significant information about Fourier components with periods less than 2DT, or no frequencies larger than 1/2DT. This gives you approximately half the number of transform points as data points. Note that the rest of the array should not be deleted, for if the autocorrelation or cross-correlation is desired, the whole array must be used when the inverse is taken.

The autocorrelation function is computed by taking the inverse transform of the power density and normalizing the result. The output is in the arrays AR (I) and AI (I) real and imaginary, respectively. The results may be printed out where indicated in the listing attached.

Times for computation: Samples are as follows:

a. Transform for 512 points: 20 secs.

b. Transform for 1024 points: 46 secs.

SUBROUTINE 'TNFRM'

SUBROUTINE 'TNFRM'

```
C*****************************************************************
C                 ** SUBROUTINE TNFRM **
C*****************************************************************
C
C      SUBROUTINE 'TNFRM' CALCULATES THE FOURIER TRANSFORM
C      USING THE COOLEY-TUKEY TRANSFORM AS WRITTEN BY
C      TRINITY UNIVERSITY  COMPUTER CENTER.
C
C
C                 ** GENERAL COMMENTS **
C*****************************************************************
C
C      THIS LISTING FOR 'TNFRM' HAS BEEN USED AS RECIEVED
C      WITHOUT MODIFICATION BY THE THE AUTHOR OF THIS
C      THESIS.
C
C
C*****************************************************************
C
C

      SUBROUTINE TNFRM
      DIMENSION ACOS(257)
      COMMON N,NM,IX,IXR,AR(1040),AI(1040),INV,INP
      COMMON LUR,LUW,IFLX,IFLY,IFLRI,IFLII,IFLRO,IFLIO,RATE
      NX1=N/2
      NX2=NX1+1
      NX3=N/4
      NX4=NX3+1
      PI2=6.2831853072
      DO 3 I=1,NX4
      IP=I-1
      ACOS(I)=COS (PI2*IP/N)
    3 CONTINUE
C     PERFORM FIRST TWO POINT TRANSFORM.
C     GO TO 40 IF INPUT DATA COMPLEX.

      IF(INP-2)30, 40, 30
   30 DO 4 J=1,NX1,1
      NXO=J+NX1
      ARQ = AR(J)
      AR(J)=AR(J)+AR(NXO)
      AR(NXO)=ARQ-AR(NXO)
    4 CONTINUE
      GO TO 56
   40 DO 50 J = 1, NX1
      NXO=J+NX1
```

SUBROUTINE 'TNFRM'

```
      ARQ=AR(J)
      AR(J)=AR(J)+AR(NXO)
      AR(NXO)=ARQ-AR(NXO)
      AIQ = AI(J)
      AI(J) = AI(J) + AI(NXO)
      AI(NXO) = AIQ - AI(NXO)
   50 CONTINUE
   56 CONTINUE
      DO 6 L=2,NM,1
      NL=NM-L
      NX=2**(NL)
      NZ=NX+NX
      NN=N-NZ+1
      DO 7 K=1,NN,NZ
      IS=K-1
      IX=IS/NX

C     CALL SUBROUTINE FOR BIT REVERSAL

      CALL REVER
      IR=IXR+1
      IF(IXR-NX3)9,9,8
    8 JH=IR-NX3
      JG=NX2-IXR
      C=-ACOS(JG)
      D=ACOS(JH)
      GO TO 10
    9 JF=NX4-IXR
      C=ACOS(IR)
      D=ACOS(JF)
      GO TO 10
   10 IF(INV-2)23, 22, 23
   22 D=-D
   23 CONTINUE
      DO 11 M=1, NX, 1
      NJ=IS+M
      NXJ=NJ+NX
      AIQ=AI(NJ)
      ARQ=AR(NJ)
      IF(INP-2) 65, 13, 65
   65 IF(L-2) 13, 12, 13
   12 CONTINUE
      AR(NJ) = AR(NXJ)*C + AR(NJ)
      AI(NJ) = AR(NXJ)*D
      AR(NXJ)=-AR(NJ) + ARQ + ARQ
      AI(NXJ)=-AI(NJ)
      GO TO 14
   13 AR(NJ) = AR(NXJ)*C - AI(NXJ)*D + AR(NJ)
      AI(NJ)=AR(NXJ)*D + AI(NXJ)*C + AI(NJ)
      AR(NXJ)=-AR(NJ)+ARQ+ARQ
```

SUBROUTINE 'TNFRM'

```
      AI(NXJ) = -AI(NJ) + AIQ + AIQ
   14 CONTINUE
   11 CONTINUE
    7 CONTINUE
    6 CONTINUE
      DO 1 I=1,N
      IX=I-1
C     CALL SUBROUTINE FOR BIT REVERSAL
      CALL REVER
      IXXR=IXR+1
      IF(IXXR-I)1,1,101
  101 CONTINUE
      ARQ=AR(IXXR)
      AR(IXXR)=AR(I)
      AR(I)=ARQ
      AIQ=AI(IXXR)
      AI(IXXR)=AI(I)
      AI(I)=AIQ
    1 CONTINUE
      IF(INV-2)70,66,70
   66 PD=1.0/N
      DO 70 I=1,N
      AR(I)=AR(I)*PD
      AI(I)=AI(I)*PD
   70 CONTINUE
      RETURN
      END
```

* SUBROUTINE REVER *

```
*  SUBROUTINE REVER *
****************************************************
*
* 'REVER' CARRIES OUT THE BIT REVERSAL OPERATION
* REQUIRED BY SUBROUTINE 'TNFRM'.
* IT WAS USED AS SUPPLIED BY THE TRINITY
* UNIVERSITY COMPUTER CENTER.
*
*
****************************************************
*
*
      ENT       REVER
REVER DC        0
      LDX   I1  /FFFE   LOAD XR1 WITH NM
      SLT       32      CLEAR A AND Q REGISTERS
      STO       REVRS
      LD    L   /FFFD   LOAD IX
      STO       NUMBR   STORE VALUE TO BE REVERSED
SHIFT LD        NUMBR
      SRT       1       SHIFT A AND Q RIGHT ONE
      STO       NUMBR
      LD        REVRS   LOAD REVERSE SUM INTO A
      SLT       1       SHIFT Q LEFT INTO A BY ONE
      STO       REVRS
      MDX    1  -1      DECREMENT XR1
      MDX       SHIFT
      STO   L   /FFFC   STORE REVERSED NO IN IXR
      BSC   I   REVER   RETURN
NUMBR DC        *-*
REVRS DC        *-*
      END
  END
```

PROGRAM 'FREQ'

```
PROGRAM 'FREQ'

C------------------------------------------------------------------
C                ** PROGRAM FREQ **
C------------------------------------------------------------------
C
C         PURPOSE
C
C            1. TO CALCULATE THE FOURIER TRANSFORM OF TIME
C            DOMAIN INPUT AND OUTPUT PULSE DATA USING THE
C            TRAPEZOIDAL OR FILON'S QUADRATURE TO EVALUATE
C            THE TRIGONOMETRIC INTEGRALS A,B,C AND D.
C            THE VALUES OF THESE INTEGRALS REPRESENT THE REAL
C            AND IMAGINARY COMPONENTS OF THE TRANSFORM.
C
C            2. TO COMBINE THE RESULTS OF THE TRANSFORMS INTO
C            AN AMPLITUDE RATIO AND PHASE LAG SUITABLE FOR
C            PLOTTING A FREQUENCY RESPONSE DIAGRAM.
C
C         PARAMETERS ENTERED VIA KEYBOARD
C
C            IFLX  - FILE NO. FOR INPUT PULSE DATA.
C            IFLY  - FILE NO. FOR OUTPUT PULSE DATA.
C            IFLA  - FILE NO. FOR VECTOR A.
C            IFLB  - FILE NO. FOR VECTOR B.
C            IFLC  - FILE NO. FOR VECTOR C.
C            IFLD  - FILE NO. FOR VECTOR D.
C            IFLPX - FILE NO. FOR REDUCED INPUT PULSE DATA.
C            IFLPY - FILE NO. FOR REDUCED OUTPUT PULSE DATA.
C            FINT  - INITIAL FREQUENCY (RAD/SEC).
C            FINC  - FREQUENCY INCREMENT (RAD/SEC).
C            FLST  - FINAL FREQUENCY (RAD/SEC).
C            ITORF - =1, TRAPEZOIDAL - =2, FILON.
C            RATE  - SAMPLE RATE USED FOR TIME DATA
C                    (POINTS/SEC).
C            IRASS - NO. OF RECORDS OF TIME DATA USED TO
C                    CALCULATE STEADY STATE.
C            IRTNF - NO. OF RECORDS OF TIME DATA USED TO
C                    CALCULATE FOURIER TRANSFORM.
C            IPICK - =1, REDUCE TIME DATA USED IN TRANSFORM
C                    BY USING ONLY EACH 'NPICK' POINT.
C                    =2 ,DO NOT REDUCE TIME DATA FOR
C                    TRANSFORM.
C            NPICK - =THE NO. OF POINTS SKIPPED FOR EACH
C                    POINT THAT IS USED WHEN IPICK=1.
C            SPECD - =0, FULL FREQ. CONTENT CURVE CALCULATED.
C                    =1, FREQ.CONTENT CURVE NORMALIZED.
C            POWR  - =1, FREQ. CONTENT CURVE CALCULATED.
C                    =2, SPECD=0, POWER SPECTRUM CALCULATED
C                    IN PLACE OF FREQ. CONTENT CURVE.
C
```

```
      PROGRAM 'FREQ'

C           ISMTH - =1, DETERMINE THE FREQ. RESPONSE DIAG.
C                   WITHOUT SPECTRAL SMOOTHING.
C                   =2, EXIT AFTER FOURIER TRANSFORM TO
C                   PERMIT SPECTRAL SMOOTHING.
C
C
C       RESTRICTIONS
C
C           1. ALL FILE NUMBERS SPECIFIED ABOVE SHOULD BE
C              DIFFERENT.
C
C           2. THE NO. OF FREQ. POINTS WILL ALWAYS BE AN
C              INTEGRAL MULTIPLE OF 'NPPR'. THE PROGRAM
C              SELECTS THE MULTIPLE SO THAT ALL THE FREQ.
C              POINTS RANGING FROM 'FINT' TO 'FLST' ARE
C              OBTAINED AND ANY EXTRA POINTS ARE CALCULATED
C              IN INCREMENTS OF 'FINC' ABOVE THE VALUE OF
C              'FLST'.
C
C           3. IF MORE THAN 200 TIME POINTS AND 120 FREQ.
C              POINTS ARE USED AND CALCULATED RESPECTIVELY,
C              THE COMPUTING TIME BECOMES EXCESSIVE.
C
C
C               ** GENERAL COMMENTS **
C-------------------------------------------------------------
C
C     THE SUBROUTINE 'ABCD1' OR ABCD2' IS CALLED TO
C     CALCULATE THE FOURIER TRANSFORM OF THE OUTPUT PULSE
C     AND THEN THE INPUT PULSE. THE TRANSFORMATION IS
C     ACHIEVED BY EVALUATING THE TRIGONOMETRIC INTEGRALS
C     'A' AND 'B' (OUTPUT) THEN 'C' AND 'D' (INPUT) FOR
C     VARIOUS VALUES OF FREQUENCY. THEREFORE 'A' BECOMES A
C     VECTOR OF FREQ. DEPENDENT VALUES REPRESENTING THE
C     REAL PART AND 'B' BECOMES A SIMILAR VECTOR
C     REPRESENTING THE IMAGINARY PART OF THE TRANSFORM FOR
C     THE OUTPUT. 'C' AND 'D' REPRESENT THE SAME VECTORS
C     RESPECTIVELY FOR THE INPUT PULSE.
C
C
C            ** TABLE OF FUNCTION OF VARIBLE NAMES **
C-------------------------------------------------------------
C
C     VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
C     NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C     OF THE APPROPRIATE SUBROUTINE.
C
C     LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C     LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
```

```
      PROGRAM 'FREQ'
C
C     NPPR=NO. OF POINTS PER RECORD IN DISK FILES.
C
C     NREC=NO. OF RECORDS USED IN IFLX AND IFLY.
C
C     DT=TIME INCREMENT BETWEEN DATA POINTS(SEC).
C
C     NPTNF=NO. TIME POINTS USED IN FOURIER TRANSFORM IF
C     THE DATA IS NOT REDUCED.
C
C     NPASS=NO. TIME POINTS USED TO DETERMINE STEADY STATE.
C
C     IRST=NO. OF FIRST RECORD WHERE DATA REDUCTION BEGINS.
C
C     IFLR IS NO. FILE CONTAINING TIME DATA USED FOURIER
C     TRANSFORM.
C
C     IFLDR IS NO. FILE CONTAINING TIME DATA FOR STEADY
C     STATE CALCULATIONS.
C
C     IFLAG INDICATES DATA TYPE, =1 OUTPUT, =2 INPUT.
C
C     SUM IS THE STEADY STATE VALUE OF THE TIME VARIABLE
C     BEFORE PULSE OCCURS.
C
C     NRF=NO. OF RECORDS OF FREQ. DATA CALCULATED.
C
C
C------------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG
      LUR=1
      LUW=2
      NPPR=40

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,36)
 36   FORMAT('THIS IS FOURIER TRANSFORM -- FREQ')

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,10)
 10   FORMAT('ENTER,IFLX,IFLY,IFLA,IFLB,IFLC,IFLD,IFLPX,',
     1'IFLPY,FINT,FINC,FLST,ITORF,RATE')
```

```
      PROGRAM 'FREQ'

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,13,0,IFLX,0,IFLY,0,IFLA,0,IFLB,0,IFLC,
     10,IFLD,0,IFLPX,0,IFLPY,1,FINT,1,FINC,1,FLST,0,ITORF,
     11,RATE,IEROR)
      WRITE(LUW,37)
 37   FORMAT('ENTER IRASS,IRTNF,IPICK,NPICK,SPECD,POWR,',
     1'ISMTH')
      CALL FFINP(LUR,7,0,IRASS,0,IRTNF,0,IPICK,0,NPICK,1,
     1SPECD,1,POWR,0,ISMTH,IEROR)

C     INITIALIZE PARAMETERS.

      NREC=IRTNF+IRASS
      DT=1.0/RATE
      NPTNF=IRTNF*NPPR
      NPASS=IRASS*NPPR
      IRST=IRASS+1

C     WRITE OUT INFORMATION.

      WRITE(LUW,33)NPASS
 33   FORMAT('NO. OF PTS USED TO CALCULATE S S=',I5)
      WRITE(LUW,34)NPTNF
 34   FORMAT('NO TIME POINTS BEFORE REDUCTION=',I5)

C     CALCULATE VECTORS A,B AND STEADY STATE (OUTPUT DATA).
C   **
C     IF IPICK=1 REDUCE NO. TIME DOMAIN DATA POINTS.

      IF(IPICK-1)60,60,61

C     CALL SUBROUTINE TO REDUCE NO. OF TIME POINTS.

 60   CALL REDPT(IFLY,IFLPY,IRTNF,NREC,NPICK,IRST)

C     SET PARAMETERS TO CALCULATE FOURIER TRANSFORM OF
C     REDUCED OUTPUT DATA.

      IFLR=IFLPY
      IFLDR=IFLY

C     SET IFL1 AND IFL2 TO FILE NOS. FOR VECTORS A AND B.

      IFL1=IFLA
      IFL2=IFLB
      GOTO 62
```

```
          PROGRAM 'FREQ'
C      SET PARAMETERS TO CALCULATE FOURIER TRANSFORM OF
C      UNREDUCED OUTPUT DATA.
 61    IFLR=IFLY
       IFLDR=IFLY
       IFL1=IFLA
       IFL2=IFLB
 62    IFLAG=1
C      SELECT QUADRATURE FORMULA
       GOTO(40,41),ITORF
C      TRAPEZOIDAL QUADRATURE.
 40    CALL ABCD1(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,NRF,SUM,
      1IPICK,IFLDR)
       GOTO 45
C      FILON'S QUADRATURE.
 41    CALL ABCD2(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,NRF,SUM,
      1IPICK,IFLDR)
C      WRITE STEADY STATE FOR OUTPUT.
 45    WRITE(LUW,11) SUM
 11    FORMAT('STEADY STATE OUTPUT=',F10.5)
C      CALCULATE VECTORS C,D AND STEADY STATE (INPUT DATA).
C    **
C      IF IPICK=1 REDUCE NO. OF DOMAIN DATA POINTS.
       IF(IPICK-1)63,63,64
C      CALL SUBROUTINE TO REDUCE TIME POINTS.
 63    CALL REDPT(IFLX,IFLPX,IRTNF,NREC,NPICK,IRST)
C      SET PARAMETERS TO CALCULATE FOURIER TRANSFORM OF
C      REDUCED INPUT DATA.
       IFLR=IFLPX
       IFLDR=IFLX
       IFL1=IFLC
       IFL2=IFLD
       GOTO 65
C      SET PARAMETERS TO CALCULATE FOURIER TRANSFORM OF
C      UNREDUCED INPUT DATA.
```

PROGRAM 'FREQ'

```
 64   IFLR=IFLX
      IFLDR=IFLX
      IFL1=IFLC
      IFL2=IFLD
 65   IFLAG=2

C     SELECT QUADRATURE FORMULA.

      GOTO(50,51),ITORF

C     TRAPEZOIDAL QUADRATURE.

 50   CALL ABCD1(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,NRF,SUM,
     1IPICK,IFLDR)
      GOTO 55

C     FILON'S QUADRATURE.

 51   CALL ABCD2(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,NRF,SUM,
     1IPICK,IFLDR)

C     WRITE STEADY STATE FOR INPUT

 55   WRITE(LUW,12) SUM
 12   FORMAT('STEADY STATE INPUT='F10.5)

C     RESET NREC AND DT TO VALUES APPLYING TO UNREDUCED
C     DATA.

      NREC=IRASS+IRTNF
      DT=DT/NPICK

C     CALL SUBROUTINE TO CALCULATE THE AREA UNDER INPUT
C     PULSE USING SIMPSON'S RULE.

      CALL SUMS(IFLX,NREC,SUM,AREA)

C     DETERMINE IF SPECTRAL SMOOTHING IS REQUIRED.

      GOTO(72,73),ISMTH

C     CALL SUBROUTINE TO CALCULATE AMP. RATIO,PHASE LAG AND
C     FREQ. CONTENT OR POWER SPECTRUM FROM DATA IN VECTORS
C     A,B,C, AND D.

 72   CALL ARPHF(IFLA,IFLB,IFLC,IFLD,NRF,AREA,FINT,FINC,
     1SPECD,POWR)
 73   CALL LINK(CHIEF)
      END
```

SUBROUTINE 'REDPT'

SUBROUTINE 'REDPT'

```
C*****************************************************************
C                ** SUBROUTINE REDPT **
C*****************************************************************
C
C      SUBROUTINE 'REDPT' REDUCES THE NO. OF TIME DOMAIN
C      DATA POINTS USED IN THE FOURIER TRANSFORM BY
C      EMPLOYING ONLY EVERY 'NPICK' POINT IN THE
C      ORIGINAL DATA.  THE SELECTED VALUES ARE STORED IN
C      A SEPARATE DATA FILE.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C*****************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      IFLR IS THE FILE CONTAINING THE UNREDUCED DATA.
C
C      IFLW=NO. OF FILE WHERE THE REDUCED DATA IS STORED.
C
C      NREC=NO. OF RECORDS USED IN UNREDUCED DATA FILE.
C
C      IRW=NO. OF RECORD IN DISK WRITE OPERATION. AT THE END
C      OF THE SUBROUTINE 'IRW' EQUALS THE TOTAL NO. OF
C      RECORDS OF REDUCED DATA.
C
C      NPICK EQUALS ONE MORE THAN THE NO. OF POINTS SKIPPED
C      BETWEEN EACH POINT SELECTED TO MAKE UP THE REDUCED
C      DATA.
C
C      IRST=NO. OF FIRST RECORD OF UNREDUCED DATA READ
C      FROM DISK.
C
C      F IS THE DIMENSIONED VARIABLE INTO WHICH UNREDUCED
C      DATA IS READ.  THE DIMENSION OF F IS SUFFICIENT TO
C      ACCOMODATE THREE RECORDS FROM DISK.
C
C      R IS THE DIMENSIONED VARIABLE INTO WHICH REDUCED
C      DATA IS SAVED.  THE DIMENSION OF R IS SUFFICIENT TO
C      ACCOMODATE ONE RECORD.
C
C      KFLAG IS A FLAG WHICH IS CHANGED FROM 1 TO 2 AFTER
C      THE LAST RECORD OF UNREDUCED DATA HAS BEEN READ
C      FROM DISK FILES.
C
C      IR3=NO. OF 3ND RECORD OF UNREDUCED DATA READ FROM
```

SUBROUTINE 'REDPT'

```
C       DISK.
C
C       L IS THE SUBSCRIPT OF THE REDUCED DATA IN CORE.
C
C       KF IS THE SUBSCRIPT FOR THE FIRST LOCATION INTO WHICH
C       A RECORD OF DATA IS READ.
C
C       KL IS THE SUBSCRIPT FOR THE LAST LOCATION INTO WHICH A
C       RECORD OF DATA IS READ.
C
C       IR4 IS THE NO. OF THE FOURTH RECORD OF UNREDUCED DATA
C       READ FROM DISK.
C
C       NSHFT IS THE SUBSCRIPT FOR THE LAST LOCATION INTO
C       WHICH THE MOST RECENT RECORD WAS READ.  ON SHIFTING
C       UNREDUCED DATA TO THE START OF THE DIMENSION, 'NSHFT'
C       IS THE LOCATION OF THE LAST POINT SHIFTED.
C
C       IS IS THE SUBSCRIPT OF FIRST UNREDUCED DATA POINT
C       SHIFTED TO START OF DIMENSION.
C
C       KS IS SUBSCRIPT FOR THE LOCATIONS INTO WHICH THE
C       SHIFTED DATA VALUES ARE READ.
C
C
C
C***************************************************************
C
C

        SUBROUTINE REDPT(IFLR,IFLW,IRTNF,IRW,NPICK,IRST)
        DIMENSION F(120),R(40)
        COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG

C       INITIALIZE PARAMETERS.

        KFLAG=1
        IR3=IRST+2
        L=0
        IRW=0
        NREC=IRTNF+IRST-1
        KL=0

C       READ TWO RECORDS OF DATA FROM DISK.

        DO 5 IRR=IRST,IR3
        KF=KL+1
        KL=KL+NPPR
        READ(IFLR'IRR)(F(LL),LL=KF,KL)
  5     CONTINUE
```

```
      SUBROUTINE 'REDPT'

      IR4=IR3+1

C     SET THE SUBSCRIPT OF THE LAST DATA POINT TO BE
C     SHIFTED BEFORE READING IN A NEW RECORD.

      NSHFT=NPPR*3
      DO 10 IRR=IR4,NREC

C     CHECK IF THERE IS ROOM IN 'F' FOR AN ADDITIONAL
C     RECORD OF UNREDUCED DATA, IE HAS 'NSHFT' DRIFTED FAR
C     ENOUGH AWAY FROM THE END OF THE DIMENSION FOR 'F' TO
C     ALLOW SPACE FOR AN ADDITIONAL RECORD. THIS OCCURS WHEN
C     'NPICK' IS A MULTIPLE OF 'NPPR'.

      IF(NSHFT-2*NPPR)61,42,42

C     CHECK IF ANOTHER RECORD OF UNREDUCED DATA SHOULD BE
C     READ FROM DISK.

 61   IF(NREC-IRR)65,66,66
 66   KS=NSHFT
      GOTO 46
 65   KFLAG=2

C     SELECT DATA POINTS IN STEPS OF NPICK STARTING AT 1.

 42   DO 11 I=1,NSHFT,NPICK
      IS=I+NPICK
      L=L+1
      R(L)=F(I)

C     CHECK FOR A FULL RECORD OF SELECTED VALUES.

      IF(L-NPPR)20,21,21

C     CHECK IF THE FINAL RECORD OF UNREDUCED DATA HAS BEEN
C     READ.

 20   IF(KFLAG-1)23,23,11

C     CHECK FOR ROOM TO READ IN A NEW RECORD OF DATA.

 23   IF(I-NPPR)11,11,44
 21   IRW=IRW+1

C     WRITE A RECORD OF SELECTED DATA ON DISK.

      WRITE(IFLW'IRW)R
      L=0
 11   CONTINUE
```

```
          SUBROUTINE 'REDPT'

      IF(KFLAG-1)45,45,71
 45   KS=0
      GOTO 46

C     SHIFT DATA TO START OF DIMENSION.

 44   KS=0

C     SHIFT REMAINING DATA TO START OF DIMENSION.

      DO 30 K=IS,NSHFT
      KS=KS+1
      F(KS)=F(K)
 30   CONTINUE

C     ADJUST THE SUBSCRIPT OF THE LAST DATA POINT SHIFTED
C     SO THAT ONLY THE DATA FROM THE NEXT RECORD READ IS
C     SHIFTED ON THE NEXT SHIFT OPERATION.

 46   NSHFT=KS+NPPR

C     DETERMINE IF ANOTHER FILE OF UNREDUCED DATA IS TO
C     READ.

      IF(NREC-IRR)40,40,41

C     SET FLAG TO INDICATE THE FINAL RECORD OF UNREDUCED
C     DATA HAS BEEN REACHED.

 40   KFLAG=2

C     READ A NEW RECORD OF UNREDUCED DATA INTO LOCATIONS
C     STARTING AT THE LOCATION IMMEDIATELY FOLLOWING THE
C     LOCATION OF THE OLDEST DATA POINT IN CORE.

 41   KF=KS+1
      KL=NSHFT
      READ(IFLR'IRR)(F(LL),LL=KF,KL)

C     IF THE LAST RECORD HAS BEEN READ RETURN TO ANALYSE
C     EACH POINT IN THAT RECORD BEFORE LEAVING THE 'IRR DO
C     LOOP'.

      IF(KFLAG-1)10,10,42
 10   CONTINUE

C     FILL THE REMAINDER OF THE FINAL RECORD OF REDUCED
C     DATA WITH THE FINAL SELECTED VALUE.
C
C     CHECK FOR AT LEAST 1/3 OF A RECORD OF PICKED VALUES
```

SUBROUTINE 'REDPT'

```
C     BEFORE FILLING THE REMAINDER.

 71   IF(L-NPPR/3)73,72,72
 72   IRW=IRW+1
      L=L+1
      DO 75 I=L,NPPR
      R(I)=R(I-1)
 75   CONTINUE

C     WRITE THE FINAL RECORD OF SELECTED DATA ON DISK.

      WRITE(IFLW'IRW)R

C     WRITE NO. OF RECORDS OF SELECTED DATA.

 73   WRITE(LUW,15)IRW
 15   FORMAT('NO. OF RECORDS OF PICKED DATA=',I3)

C     MODIFY DT FOR REDUCED NO. OF DATA POINTS.

      DT=1.0/RATE*NPICK

C     WRITE OUT THE MODIFIED VALUE OF DT.

      WRITE(LUW,16)DT
 16   FORMAT('TIME INC.=',F7.4,1X,'SEC')
      RETURN
      END
```

SUBROUTINE 'ABCD1'

```
SUBROUTINE 'ABCD1'

C***********************************************************************
C                 ** SUBROUTINE ABCD1 **
C***********************************************************************
C
C      SUBROUTINE ABCD1 USES THE TRAPEZOIDAL QUADRATURE TO
C      EVALUATE THE A,B,C AND D TRIGONOMETRIC INTEGRALS
C      OF THE FOURIER TRANSFORM OVER A RANGE OF FREQUENCIES.
C
C
C            ** TABLE OF FUNCTION OF VARIBLE NAMES **
C***********************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      X IS A DIMENSIONED VARIABLE INTO WHICH THE TIME DOMAIN
C      PULSE DATA IS READ.
C
C      A IS A DIMENSIONED VARIABLE IN WHICH THE REAL PART OF
C      THE FOURIER TRANSFORM IS STORED (FREQ. DATA).
C
C      B IS A DIMENSIONED VARIABLE IN WHICH THE IMAG. PART OF
C      THE FOURIER TRANS FORM IS STORED (FREQ. DATA).
C
C      IFLR=NO. OF FILE CONTAINING THE TIME DOMAIN PULSE
C      DATA USED (INPUT OR OUTPUT).
C
C      IFL1=NO. OF THE FILE WHERE THE VALUES OF THE COSINE
C      PRODUCT INTEGRAL ARE TO BE STORED.
C
C      IFL2=NO. OF THE FILE WHERE THE VALUES OF THE SINE
C      PRODUCT INTEGRAL ARE TO BE STORED.
C
C      NF=NO. OF FREQ. POINTS CALCULATED, IT MUST BE AN
C      INTEGRAL MULTIPLE OF NPPR.
C
C      NFR=NO. OF RECORDS OF FREQUENCY DATA.
C
C      W=FREQUENCY (RAD/SEC).
C
C      SUM=THE AVERAGE STEADY STATE OF OF TIME DATA.
C
C      IRW=RECORD NUMBER IN DISK WRITE OPERATIONS.
C
C      IRASS=NO. OF RECORDS USED TO CALCULATE STEADY STATE.
C      IRR=RECORD NO. IN DISK READ OPERATIONS.
C
```

SUBROUTINE 'ABCD1'

```
C     KL IS SUBSCRIPT FOR LAST LOCATION INTO WHICH A RECORD
C     OF DATA IS READ.
C
C     NRTNF=NO. OF FIRST RECORD OF TIME DATA THAT IS TO BE
C     USED IN FOURIER TRANSFORM.
C
C     NT=TOTAL NO. OF TIME POINTS USED IN FOURIER TRANS.
C
C     KF IS LIKE KL BUT REFERS TO FIRST LOCATION.
C
C     L*DT GIVES THE TOTAL TIME ASSOCIATED WITH EACH TIME
C     DATA POINT.
C
C     K IS THE SUBSCRIPT FOR THE RECORD OF FREQ. DATA IN
C     CORE.
C
C     SUMB IS AN INTERMEDIATE VALUE OF SINE INTEGRAL.
C
C     SUMA IS AN INTERMEDIATE VALUE OF COSINE INTEGRAL.
C
C     KEC IS SUBSCRIPT OF LOCATION WHERE ERROR OF CLOSURE
C     CALCULATIONS BEGIN.
C
C     ERCL=ERROR OF CLOSURE ON THE TAIL OF THE TIME PULSE.
C
C
C*******************************************************************
C
C

      SUBROUTINE ABCD1(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,
     1NRF,SUM,IPICK,IFLDR)
      DIMENSIONX(600),A(400),B(400)
      COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG

C     CALCULATE NO. OF FREQUENCY POINTS REQUIRED.

      NF=(FLST-FINT)/FINC+1

C     CALCULATE NO. OF RECORDS TO BE USED IN IFL1 AND IFL2.

      NRF=NF/NPPR+1

C     DETERMINE NO. OF FREQUENCY POINTS WHICH WILL BE
C     CALCULATED (INTEGER MULTIPLE OF NPPR).

      NF=NPPR*NRF

C     WRITE NO. OF FREQUENCY POINTS CALCULATED.
```

SUBROUTINE 'ABCD1'

```
      WRITE(LUW,17) NF
 17   FORMAT('NO OF PTS FREQ DATA CALCULATED=',I5)
C     SET INITIAL FREQUENCY.
      W=FINT
      SUM=0.0
      IRW=0
C     CALCULATE STEADY STATE.
      DO 9 IRR=1,IRASS
C     READ A RECORD OF TIME DATA FROM DISK.
      READ (IFLDR'IRR) X
C     SUM THE VALUES OVER 'IRASS' RECORDS.
      DO 10 I=1,NPPR
      SUM=SUM+X(I)
 10   CONTINUE
 9    CONTINUE
C     AVERAGE SUM TO OBTAIN STEADY STATE.
      SUM=SUM/(NPPR*IRASS)
      KL=0
C     CHECK IF TIME DATA HAS BEEN REDUCED OR NOT.
      IF(IPICK-1)60,60,61
 60   NRTNF=1
      NT=NREC*NPPR
      GOTO 62
 61   NRTNF=IRASS+1
      NT=NPPR*(NREC-IRASS)
C     WRITE TOTAL NO. OF TIME POINTS USED IN TRANSFORM.
 62   WRITE(LUW,18)NT
 18   FORMAT('NO TIME POINTS USED AFTER PICK=',I5)
C     READ ALL OF TIME DATA FROM DISK.
      DO 100 IRR=NRTNF,NREC
      KF=KL+1
      KL=KF+NPPR-1
      READ(IFLR'IRR)(X(LL),LL=KF,KL)
 100  CONTINUE
```

```
      SUBROUTINE 'ABCD1'

C     BEGIN EVALUATION OF TRIGONOMETRIC INTEGRALS.

      DO 102 K=1,NF

C     INITIALIZE PARAMETERS FOR EACH FREQUENCY.

      L=0
      SUMA=0.0
      SUMB=0.0
      DO 103 J=1,NT
      SUMA=SUMA+(X(J)-SUM)*COS(W*L*DT)
      SUMB=SUMB+(X(J)-SUM)*SIN(W*L*DT)
      L=L+1
 103  CONTINUE

C     SET A AND B EQUAL THE VALUE OF INTEGRALS FOR ONE FREQ.

      A(K)=SUMA*DT
      B(K)=SUMB*DT

C     INCREMENT FREQUENCY.

      W=W+FINC
 102  CONTINUE

C     WRITE THE VECTORS OF FREQUENCY DATA ON DISK.

      KL=0
      DO 105 IRW=1,NRF
      KF=KL+1
      KL=KF+NPPR-1
      WRITE(IFL1'IRW)(A(LL),LL=KF,KL)
      WRITE(IFL2'IRW)(B(LL),LL=KF,KL)
 105  CONTINUE

C     DETERMINE IF INPUT OR OUTPUT DATA WAS PROCESSED.

      IF(IFLAG-1)205,205,206
 205  WRITE(LUW,200)
 200  FORMAT('ERROR OF CLOSURE OUTPUT')
      GOTO 207
 206  WRITE(LUW,201)
 201  FORMAT('ERROR OF CLOSURE INPUT')

C     CALCULATE THE ERROR OF CLOSURE FOR EVERY THIRD POINT
C     IN THE LAST RECORD OF TIME DATA PROCESSED.

 207  KEC=NT-NPPR+1
      DO 107 I=KSS,NT,3
```

SUBROUTINE 'ABCD1'

```
      ERCL=X(I)-SUM
C     WRITE ERROR OF CLOSURE.
      WRITE(LUW,79)ERCL
 79   FORMAT(F10.5)
 107  CONTINUE
      RETURN
      END
```

SUBROUTINE 'ABCD2'

SUBROUTINE 'ABCD2'

```
C
C**************************************************************
C                 ** SUBROUTINE ABCD2 **
C**************************************************************
C
C     SUBROUTINE ABCD2 USES FILON'S QUADRATURE TO EVALUATE
C     THE A,B,C AND D TRIGONOMETRIC INTEGRALS OF THE FOURIER
C     TRANSFORM OVER A RANGE OF FREQUENCIES.
C
C
C                 ** GENERAL COMMENTS **
C**************************************************************
C
C     THE ODD AND EVEN SUMS DEFINED IN THIS PROGRAM ARE THE
C     REVERSE OF THOSE GIVEN IN THE REFERENCE FOR FILON'S
C     QUADRATURE, BECAUSE THE FIRST SUBSCRIPT IN THIS
C     PROGRAM IS 1 RATHER THAN 0.
C
C
C          ** TABLE OF FUNCTION OF VARIABLE NAMES **
C**************************************************************
C
C     VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C     ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C     THAT OF THE MAINLINE.
C
C
C     X IS A DIMENSIONED VARIABLE INTO WHICH EACH RECORD OF
C     TIME DOMAIN PULSE DATA IS READ.
C
C     A IS A DIMENSIONED VARIABLE IN WHICH EACH RECORD OF
C     THE REAL PART OF THE FOURIER TRANSFORM IS STORED
C     (FREQ. DATA).
C
C     B IS A DIMENSIONED VARIABLE IN WHICH EACH RECORD OF
C     THE IMAG. PART OF THE FOURIER TRANSFORM IS STORED
C     (FREQ. DATA).
C
C     IFLR=NO. OF FILE CONTAINING THE TIME DOMAIN PULSE
C     DATA USED (INPUT OR OUTPUT).
C
C     IFL1=NO. OF THE FILE WHERE THE VALUES OF THE COSINE
C     PRODUCT INTEGRAL ARE TO BE STORED.
C
C     IFL2=NO. OF THE FILE WHERE THE VALUES OF THE SINE
C     PRODUCT INTEGRAL ARE TO BE STORED.
C
C     NF=NO. OF FREQ. POINTS CALCULATED, IT MUST BE AN
```

SUBROUTINE 'ABCD2'

```
C     INTEGRAL MULTIPLE OF NPPR.
C
C     NFR=NO. OF RECORDS OF FREQUENCY DATA.
C
C     W=FREQUENCY (RAD/SEC).
C
C     SUM=THE AVERAGE STEADY STATE OF OF TIME DATA.
C
C     IRW=RECORD NUMBER IN DISK WRITE OPERATIONS.
C
C     IRASS=NO. OF RECORDS USED TO CALCULATE STEADY STATE.
C
C     IRR=RECORD NO. IN DISK READ OPERATIONS.
C
C     ODD SUM IMPLIES SUM OF POINTS 1,3,5,---2N-1.
C
C     EVEN SUM IMPLIES SUM OF POINTS 2,4,6,---2N.
C
C     SUMEC IMPLIES THE EVEN SUM OF COSINE PRODUCT VALUES.
C
C     SUMES IMPLIES THE EVEN SUM OF SINE PRODUCT VALUES.
C
C     SUMOC IMPLIES THE ODD SUM OF COSINE PRODUCT VALUES.
C
C     SUMOS IMPLIES THE ODD SUM OF SINE PRODUCT VALUES.
C
C     L*DT GIVES TOTAL THE TIME ASSOCIATED WITH EACH ODD
C     TIME POINT.
C
C     M*DT GIVES THE TOTAL TIME ASSOCIATED WITH EACH EVEN
C     TIME POINT.
C
C
C     NRTNF=THE NO. OF THE FIRST RECORD OF TIME DATA USED
C     FOURIER TRANSFORM CALCULATIONS.
C
C     NR=NO. OF SECOND LAST RECORD OF TIME DATA USED IN
C     FOURIER TRANSFORM CALCULATIONS.
C
C     K IS THE SUBSCRIPT FOR THE RECORD OF FREQUENCY DATA
C     CURRENTLY IN CORE.
C
C     NP=LAST POINT,IN EACH RECORD OF TIME DATA, USED IN
C     THE CALCULATION OF ODD SUM.
C
C     NEPLR=LAST POINT OF TIME DATA USED IN CALCULATION OF
C     EVEN SUM.
C
C     FAC IS THE VALUE OF THE PRODUCT OF THE FIRST TIME
C     POINT AND THE COSINE FUNCTION.
```

SUBROUTINE 'ABCD2'

```
C
C     FAS IS THE VALUE OF THE PRODUCT OF THE FIRST TIME
C     POINT AND THE SINE FUNCTION.
C
C     IRTNF=NO. OF 2ND REC.OF TIME DATA USED IN TFANSFORM.
C
C     FBC IS THE VALUE OF THE PRODUCT OF THE LAST TIME
C     POINT AND THE COSINE FUNCTION.
C
C     FBS IS THE VALUE OF THE PRODUCT OF THE LAST TIME
C     POINT AND THE SINE FUNCTION.
C
C     ALP, BET AND GAM DENOTE THE ALPHA,BETA AND GAMMA
C     DEFINED IN THE REFERENCE FOR FILON'S QUADRATURE.
C
C     ERCL=ERROR OF CLOSURE ON THE TAIL OF THE TIME PULSE.
C
C
C*************************************************************
C
C

      SUBROUTINE ABCD2(FINT,FINC,FLST,IFLR,IFL1,IFL2,NREC,
     1NRF,SUM,IPICK,IFLDR)
      DIMENSION X(40),A(40),B(40)
      COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG

C     CALCULATE NO. OF FREQUENCY POINTS REQUIRED.

      NF=(FLST-FINT)/FINC+1

C     CALCULATE NO. OF RECORDS TO BE USED IN IFL1 AND IFL2.

      NRF=NF/NPPR+1

C     DETERMINE NO. OF FREQ. POINTS WHICH WILL BE CALCULATED
C     (INTEGER MULTIPLE OF NPPR).

      NF=NPPR*NRF

C     WRITE NO. OF FREQ. POINTS CALCULATED.

      WRITE(LUW,17) NF
 17   FORMAT('NO OF PTS FREQ DATA CALCULATED=',I5)
      NT=NPPR*NREC

C     WRITE TOTAL NO. OF TIME POINTS USED IN TRANSFORM.

      WRITE(LUW,18)NT
 18   FORMAT('NO. OF TIME POINTS AFTER PICK=',I5)
```

```
      SUBROUTINE 'ABCD2'

C     INITIALIZE FREQUENCY.

      W=FINT
      SUM=0.0
      IRW=0

C     CALCULATE STEADY STATE.

      DO 9 IRR=1,IRASS

C     READ A RECORD OF TIME DATA FROM DISK.

      READ(IFLDR'IRR) X

C     SUM THE VALUES OVER 'IRASS' RECORDS.

      DO 10 I=1,NPPR
      SUM=SUM+X(I)
 10   CONTINUE
 9    CONTINUE

C     AVERAGE SUM TO OBTAIN STEADY STATE.

      SUM=SUM/(NPPR*IRASS)

C     CHECK IF TIME DATA HAS BEEN REDUCED OR NOT.

      IF(IPICK-1)60,60,61

C     INITIALIZE PARAMETERS FOR EVALUATION OF INTEGRALS.

 60   NRTNF=1
      GOTO 62
 61   NRTNF=IRASS+1
 62   NR=NREC-1
      K=0
      NP=NPPR-1
      NEPLR=NPPR-2

C     BEGIN EVALUATION OF TRIGONOMETERIC INTEGRALS.

      DO 15 KW=1,NF

C     INITIALIZE PARAMETERS FOR EACH FREQUENCY.

      SUMES=0.0
      SUMOS=0.0
      SUMEC=0.0
      SUMOC=0.0
```

```
      SUBROUTINE 'ABCD2'

      L=0
      M=1

C     READ FIRST RECORD OF TIME DATA USED IN TRANSFORM.

      READ(IFLR'NRTNF)X

C     CALCULATE THE ODD SUM FOR THE FIRST RECORD OF TIME
C     DATA.

      DO 20 I=1,NP,2

C     CHECK FOR THE FIRST POINT IN ODD SUM.

      IF(I-1) 30,30,31
 30   SUMOC=(X(I)-SUM)*COS(W*L*DT)/2.0
      FAC=SUMOC
      SUMOS=(X(I)-SUM)*SIN(W*L*DT)/2.0
      FAS=SUMOS
      GOTO 21
 31   SUMOC=SUMOC+(X(I)-SUM)*COS(W*L*DT)
      SUMOS=SUMOS+(X(I)-SUM)*SIN(W*L*DT)
 21   L=L+2
 20   CONTINUE

C     CALCULATE THE EVEN SUM FOR THE FIRST RECORD OF TIME
C     DATA.

      DO 22 J=2,NPPR,2
      SUMEC=SUMEC+(X(J)-SUM)*COS(W*M*DT)
      SUMES=SUMES+(X(J)-SUM)*SIN(W*M*DT)
      M=M+2
 22   CONTINUE

C     CONTINUE TO CALCULATE EVEN AND ODD SUMS FROM THE 2ND
C     TO THE 2ND LAST RECORD OF TIME DATA USED IN TRANSFORM.

      IRTNF=NRTNF+1
      DO 25 IRR=IRTNF,NR

C     READ A RECORD OF TIME DATA.

      READ(IFLR'IRR)X

C     CALCULATE THE ODD SUM FOR THE RECORD.

      DO 26 I=1,NP,2
      SUMOC=SUMOC+(X(I)-SUM)*COS(W*L*DT)
      SUMOS=SUMOS+(X(I)-SUM)*SIN(W*L*DT)
      L=L+2
```

SUBROUTINE 'ABCD2'

```
 26    CONTINUE

C      CALCULATE THE EVEN SUM FOR THE RECORD.

       DO 27 J=2,NPPR,2
       SUMEC=SUMEC+(X(J)-SUM)*COS(W*M*DT)
       SUMES=SUMES+(X(J)-SUM)*SIN(W*M*DT)
       M=M+2
 27    CONTINUE
 25    CONTINUE

C      CALCULATE EVEN AND ODD SUM FOR LAST RECORD OF TIME
C      DATA USED IN TRANSFORM.
C    **
C      READ THE LAST RECORD OF TIME DATA.

       READ(IFLR'NREC)X

C      CALCULATE THE ODD SUM FOR THE LAST RECORD.

       DO 28 I=1,NP,2

C      CHECK FOR THE 2ND LAST POINT IN THE 2ND LAST RECORD.

       IF(I-NP)45,46,46
 46    FBC=(X(I)-SUM)*COS(W*L*DT)/2.0
       SUMOC=SUMOC+FBC
       FBS=(X(I)-SUM)*SIN(W*L*DT)/2.0
       SUMOS=SUMOS+FBS
       GOTO 24
 45    SUMOC=SUMOC+(X(I)-SUM)*COS(W*L*DT)
       SUMOS=SUMOS+(X(I)-SUM)*SIN(W*L*DT)
 24    L=L+2
 28    CONTINUE

C      CALCULATE THE EVEN SUM FOR THE LAST RECORD.

       DO 29 J=2,NEPLR,2
       SUMEC=SUMEC+(X(J)-SUM)*COS(W*M*DT)
       SUMES=SUMES+(X(J)-SUM)*SIN(W*M*DT)
       M=M+2
 29    CONTINUE

C      EVALUATE ALP,BET,GAM.

       R=W*DT

C      IF R IS LESS THAN 0.35 RADIANS EVALUATE ALP,BET AND
C      GAM BY A SERIES APPROXIMATION.
```

SUBROUTINE 'ABCD2'

```
      IF (R-0.35) 35,35,36
 35   ALP=2.0/45.0*R**3.-2.0/315.0*R**5.+2.0/4725.0*R**7.
      BET=2.0/3.0+2.0/15.0*R**2.-4.0/105.0*R**4.+2.0/567.0*
     1R**6.-4.0/22275.0*R**8.
      GAM=4.0/3.0-2.0/15.0*R**2.+R**4./270.0-R**6./11340.0+
     1R**8./997920.0
      GOTO 38
 36   ALP=1.0/R+SIN(2.0*R)/(2.0*R**2.)-2.0*(SIN(R))**2./R*
     1*3.
      BET=2.0*(((COS(R))**2.+1.0)/R**2.-SIN(2.0*R)/R**3.)
      GAM=4.0*(SIN(R)/R**3.-COS(R)/R**2.)
 38   K=K+1

C     SET A AND B EQUAL TO THE VALUES OF THE COSINE AND SINE
C     INTEGRALS RESPECTIVELY.

      A(K)=DT*(ALP*(FBS-FAS)+BET*SUMOC+GAM*SUMEC)
      B(K)=DT*(ALP*(FAC-FBC)+BET*SUMOS+GAM*SUMES)

C     CHECK IF A FULL RECORD OF FREQ. DATA HAS BEEN
C     CALCULATED.

      IF(K-NPPR)13,51,51
 51   IRW=IRW+1

C   **
C     WRITE A RECORD OF FREQ. DATA FOR COSINE AND SINE
C     INTEGRALS RESPECTIVELY.

      WRITE(IFL1'IRW) A
      WRITE(IFL2'IRW) B

C     RESET SUBSCRIPT FOR FREQ. DATA.

      K=0

C     INCREMENT FREQUENCY.

 13   W=W+FINC
 15   CONTINUE

C     CALCULATE THE ERROR OF CLOSURE FOR EVERY THIRD POINT
C     IN THE LAST RECORD OF TIME DATA USED IN TRANSFORM.

      WRITE(LUW,78)
 78   FORMAT('ERROR OF CLOSURE EVERY 3RD PT.IN LAST REC.')
      DO 77 I=1,NPPR,3
      ERCL=X(I)-SUM

C     WRITE ERROR OF CLOSURE.
```

SUBROUTINE 'ABCD2'

```
      WRITE(LUW,79) ERCL
79    FORMAT(2X,F10.5)
77    CONTINUE
      RETURN
      END
```

SUBROUTINE 'SUMS'

SUBROUTINE 'SUMS'

```
C*****************************************************************
C                ** SUBROUTINE SUMS **
C*****************************************************************
C
C      SUBROUTINE SUMS CALCULATES THE AREA UNDER THE CURVE OF
C      THE INPUT PULSE USING THE SIMPSON'S QUADRATURE.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C*****************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      V IS THE DIMENSIONED VARIABLE INTO WHICH EACH RECORD
C      OF DATA IS READ DURING THE CALCULATION OF AREA.
C
C      IRTNF AT FIRST =NO. OF FIRST RECORD OF TIME DATA USED
C      IN CALCULATION, LATER IT IS SET EQUAL TO NO. OF
C      SECOND RECORD.
C
C      AREA=AREA UNDER THE CURVE AT EACH STAGE IN THE
C      CALCULATION, IE. =TOTAL AREA AT END OF SUBROUTINE.
C
C      N2=NO. OF LAST POINT CONSIDERED IN A RECORD, IT IS
C      CHANGED AS REQUIRED.
C
C      NSL=NO. OF SECOND LAST RECORD USED IN CALCULATION OF
C      AREA.
C
C
C*****************************************************************
C
C

      SUBROUTINE SUMS(IFLX,NREC,SUM,AREA)
      DIMENSION V(40)
      COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG
      IRTNF=IRASS+1

C     READ FIRST RECORD OF DATA.

      READ(IFLX'IRTNF)V

C     SET AREA EQUAL TO THE VALUE OF THE FIRST POINT.

      AREA=V(1)-SUM
```

```
      SUBROUTINE 'SUMS'

C     MULTIPLY EVERY 2ND POINT BY 4 STARTING WITH 2ND
C     POINT IN RECORD.

      DO 11 I=2,NPPR,2
      AREA=AREA+(V(I)-SUM)*4.0
 11   CONTINUE
      N2=NPPR-1

C     MULTIPLY EVERY 2ND POINT BY 2 STARTING WITH 3RD POINT.

      DO 13 I=3,N2,2
      AREA=AREA+(V(I)-SUM)*2.0
 13   CONTINUE
      NSL=NREC-1
      IRTNF=IRTNF+1

C     READ FROM 2ND TO 2ND LAST RECORD OF DATA.

      DO 14 IRR=IRTNF,NSL
      READ(IFLX'IRR)V

C     MULTIPLY EVERY 2ND POINT BY 2 STARTING WITH 1ST POINT
C     IN RECORD.

      DO 15 I=1,N2,2
      AREA=AREA+(V(I)-SUM)*2.0
 15   CONTINUE

C     MULTIPLY EVERY 2ND POINT BY 4 STARTING WITH 2ND POINT
C     IN RECORD.

      DO 16 I=2,NPPR,2
      AREA=AREA+(V(I)-SUM)*4.0
 16   CONTINUE
 14   CONTINUE

C     READ THE LAST RECORD OF DATA.

      READ(IFLX'NREC)V
      N2=N2-2

C     MULTIPLY EVERY 2ND POINT BY 2 FROM THE 1ST POINT UP
C     TO INCLUDE 4TH FROM LAST POINT.

      DO 17 I=1,N2,2
      AREA=AREA+(V(I)-SUM)*2.0
 17   CONTINUE
      N2=N2+1
```

```
      SUBROUTINE 'SUMS'

C     MULTIPLY EVERY 2ND POINT BY 4 FROM THE 2ND POINT UP
C     TO INCLUDE 3RD FROM LAST POINT.

      DO 18 I=2,N2,2
      AREA=AREA+(V(I)-SUM)*4.0
 18   CONTINUE
      N2=N2+1

C     ADD VALUE OF 2ND LAST POINT TO AREA.

      AREA=AREA+(V(N2)-SUM)

C     THE LAST POINT IN THE LAST RECORD IS NEGLECTED
C     BECAUSE SIMPSON'S RULE REQUIRES AN EVEN NO. OF
C     INTERVALS, ALSO, IF THE PULSE IS CLOSED, THE LAST
C     VALUE WILL BE ZERO SO NO ERROR WILL BE INCURRED.
C   **
C     CALCULATE THE TOTAL AREA.

      AREA=AREA*DT/3.0

C     WRITE OUT VALUE OF AREA.

      WRITE(LUW,40) AREA
 40   FORMAT('AREA UNDER INPUT CURVE=',F10.5)
      RETURN
      END
```

SUBROUTINE 'ARPHF'

```
SUBROUTINE 'ARPHF'

C*************************************************************
C                ** SUBROUTINE ARPHF **
C*************************************************************
C
C      SUBROUTINE 'ARPHF' USES THE REAL AND IMAGINARY
C      COMPONENTS OF THE FOURIER TRANSFORM OF THE INPUT AND
C      THE OUTPUT PULSES, AS CALCULATED BY SUBROUTINE ABCD1
C      OR ABCD2, TO CALCULATE THE AMPLITUDE RATIO, PHASE LAG
C      AND FREQUENCY CONTENT FOR THE SYSTEM TESTED.
C
C
C                ** GENERAL COMMENTS **
C*************************************************************
C
C      THE DIMENSIONED VARIABLES A,B,C,D INITIALLY
C      REPRESENT THE REAL AND IMAGINARY PARTS OF THE FOURIER
C      TRANSFORM OF THE OUTPUT AND INPUT PULSES RESPECTIVELY.
C      AFTER A RECORD OF EACH VARIABLE IS READ FROM DISK
C      INTO CORE, THE VALUES IN EACH LOCATION ARE OVERLAID
C      WITH THE VALUES OF THE AMPLITUDE RATIO, PHASE LAG AND
C      FREQUENCY CONTENT.  THESE VALUES ARE THEN WRITTEN ON
C      TO DISK IN THE SAME AREA THAT A,B,C,D WERE READ FROM.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C*************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C
C      A=REAL PART FOURIER TRANSFORM OUTPUT. (INITIALLY)
C
C      B=IMAG. PART FOURIER TRANSFORM OUTPUT. (INITIALLY)
C
C      C=REAL PART FOURIER TRANSFORM INPUT. (INITIALLY)
C
C      D=IMAG. PART FOURIER TRANSFORM INPUT. (INITIALLY)
C
C      IRR=THE NO. OF THE RECORD IN DISK READ WRITE
C      OPERATIONS.
C
C      W=FREQUENCY (RAD/SEC).
C
C      Z IS THE DENOMINATOR TERM USED IN CALCULATION OF THE
C      REAL AND IMAGINARY PARTS OF THE TRANSFER FUNCTION.
C
C      RE=REAL PART OF TRANSFER FUNCTION.
C
```

SUBROUTINE 'ARPHF'

```
C      CO=IMAGINARY PART OF TRANSFER FUNCTION.
C
C      AG=PHASE ANGLE.
C
C
C*****************************************************************
C
C

       SUBROUTINE ARPHF(IFLA,IFLB,IFLC,IFLD,NRF,AREA,FINT,
      1FINC,SPECD,POWR)
       DIMENSION A(40),B(40),C(40),D(40)
       COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG

C      INITIALIZE FREQUENCY.

       W=FINT

C      INITIALIZE PARAMETERS TO CALCULATE ANGLE.

       ADJ=0.0
       BIT0=1.0
       STORE=1.0
       BIT2=1.0

C      READ ONE RECORD FROM EACH FILE OF FREQ. DATA.

       DO 10 IRR=1,NRF
       READ(IFLA'IRR)A
       READ(IFLB'IRR)B
       READ(IFLC'IRR)C
       READ(IFLD'IRR)D
       DO 11 I=1,NPPR
       Z=C(I)**2.+D(I)**2.

C      CALCULATE REAL AND IMAGINARY PARTS OF THE TRANSFER
C      FUNCTION.

       RE=(A(I)*C(I)+B(I)*D(I))/Z
       CO=(A(I)*D(I)-B(I)*C(I))/Z

C      CALCULATE AMPLITUDE RATIO.

       A(I)=ABS((RE**2.+CO**2.)**0.5)

C      CALL A SUBROUTINE TO CALCULATE THE PHASE ANGLE.

       CALL ANGLE(CO,RE,AG,ADJ,BIT2,IRR,I,BIT0,STORE)
       B(I)=AG
```

```
      SUBROUTINE 'ARPHF'

C     CALCULATE THE FREQUENCY CONTENT OR THE POWER
C     SPECTRUM.

      C(I)=(ABS(C(I)**2+D(I)**2)**0.5)**POWR/(AREA**SPECD)

C     CALCULATE FREQUENCY.

      D(I)=W
      W=W+FINC
 11   CONTINUE

C     WRITE ONE RECORD OF OF AMP. RATIO DATA ON DISK.

      WRITE(IFLA'IRR)A

C     WRITE ONE RECORD OF PHASE ANGLE DATA ON DISK.

      WRITE(IFLB'IRR)B

C     WRITE ONE RECORD OF OF FREQ. CONTENT OR POWER SPECTRUM
C     DATA ON DISK.

      WRITE(IFLC'IRR)C

C     WRITE ONE RECORD OF OF FREQ. VALUES ON DISK.

      WRITE(IFLD'IRR)D
 10   CONTINUE

C     WRITE OUT THE FILE NUMBERS WHERE THE VARIOUS PIECES
C     OF THE INFORMATION ARE STORED.

      WRITE(LUW,20)IFLA
 20   FORMAT('AMP RATIO IN FILE',I3)
      WRITE(LUW,21)IFLB
 21   FORMAT('PHAS LAG IN FILE',I3)
      WRITE(LUW,22)IFLC
 22   FORMAT('FREQ CONT IN FILE',I3)
      WRITE(LUW,23)IFLD
 23   FORMAT('FREQ VALUES IN FILE',I3)
      RETURN
      END
```

SUBROUTINE 'ANGLE'

SUBROUTINE 'ANGLE'

```
C*************************************************************
C                  ** SUBROUTINE ANGLE **
C*************************************************************
C
C      SUBROUTINE 'ANGLE' UTILIZES THE FORTRAN SUPPLIED
C      SUBPROGRAM 'ATAN' TO EVALUATE THE PHASE ANGLE OF THE
C      TRANSFER FUNCTION.  IT IS DESIGNED TO SAVE THE STATUS
C      OF THE LAST ANGLE VALUE, SO IF IT IS CALLED FOR
C      SUCCESSIVE VALUES OF THE REAL AND IMAGINARY
C      COMPONENTS OF THE TRANSFER FUNCTION, AN ACCUMULATIVE
C      ANGLE UP TO SEVERAL THOUSAND DEGREES MAY BE
C      DETERMINED.
C
C
C                  ** GENERAL COMMENTS **
C*************************************************************
C
C      THE SUBPROGRAM 'ATAN' EVALUATES THE ANGLE BETWEEN
C      + AND - 90 DEGREES (THE REGION NEAR ZERO FOR WHICH
C      THE TANGENT OF AN ANGLE IS DEFINED).  THEREFORE, TO
C      OBTAIN AN ACCUMULATIVE PHASE ANGLE, THIS SUBROUTINE
C      ADDS + OR- 180 DEGREES TO THE 'ATAN' ANGLE EACH TIME
C      'Q' CHANGES SIGN.  THE SUBROUTINE ASSUMES THAT IF 'Q'
C      CHANGES SIGN WHEN THE PREVIOUS ATAN ANGLE WAS IN THE
C      UPPER R.H.QUADRANT THEN THE NEW ADJUSTED ANGLE LIES
C      IN THE UPPER L.H. QUADRANT. SIMILARILY FROM LOWER R.H.
C      TO LOWER L.H. QUADRANT. THEREFORE LARGE CHANGES IN
C      ANGLE CAN CAUSE THE SUBROUTINE TO ERR.
C
C
C              ** TABLE OF FUNCTION OF VARIABLE NAMES **
C*************************************************************
C
C      VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
C      ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
C      THAT OF THE MAINLINE.
C
C      P=IMAGINARY PART OF TRANSFER FUNCTION.
C
C      Q=REAL PART OF TRANSFER FUNCTION.
C
C      A=VALUE OF ARCTAN FUNCTION AT BEGINNING OF
C      SUBROUTINE, BUT = ACCUMULATIVE ANGLE WHEN RETURNED
C      IN ARGUMENT LIST.
C
C      ADJ=ADJUSTMENT FOR ACCUMULATIVE ANGLE.
C
C      BIT2=-1.0 IF ANGLE IS DECREASING, =+1.0 IF ANGLE IS
```

SUBROUTINE 'ANGLE'

[illegible]UTINE 'ANGLE'

```
************************************************************
          ** SUBROUTINE 'ANGLE' **
************************************************************

   SUBROUTINE 'ANGLE' UTILIZES THE FORTRAN SUPPLIED
   SUBPROGRAM 'ATAN' TO EVALUATE THE PHASE ANGLE OF THE
   TRANSFER FUNCTION.  IT IS DESIGNED TO SAVE THE STATUS
   OF THE LAST ANGLE VALUE, SO IF IT IS CALLED FOR
   SUCCESSIVE VALUES OF THE REAL AND IMAGINARY
   COMPONENTS OF THE TRANSFER FUNCTION, AN ACCUMULATIVE
   ANGLE UP TO SEVERAL THOUSAND DEGREES MAY BE
   DETERMINED.


          ** GENERAL COMMENTS **
************************************************************

   THE SUBPROGRAM 'ATAN' EVALUATES THE ANGLE [illegible]
   + AND - 90 DEGREES (THE REGION [illegible]
   THE TANGENT OF AN ANGLE IS DEFINED).  THEREFORE, TO
   OBTAIN AN ACCUMULATIVE PHASE ANGLE, THIS SUBROUTINE
   ADDS + OR - 180 DEGREES TO THE 'ATAN' [illegible]
   'Q' CHANGES SIGN.  THE SUBROUTINE [illegible] THAT IF 'Q'
   CHANGES SIGN WHEN THE PREVIOUS ATAN ANGLE WAS IN THE
   UPPER R.H. QUADRANT THEN THE NEW ADJUSTED ANGLE L[illegible]
   IN THE UPPER L.H. QUADRANT.  SIMILARLY [illegible]
   TO LOWER L.H. QUADRANT.  THEREFORE LARGE [illegible]
   ANGLE CAN CAUSE THE SUBROUTINE TO [illegible].


          ** TABLE OF FUNCTION OF VARIABLES [illegible] **
************************************************************

   VARIABLES IN COMMON AND IN THE ARGUMENT LIST WHICH
   ARE NOT DESCRIBED IN THIS LISTING ARE DESCRIBED IN
   THAT OF THE MAINLINE.

   [illegible]=IMAGINARY PART OF TRANSFER FUNCTION.

   [illegible]REAL PART OF TRANSFER FUNCTION.

   [illegible]VALUE OF ARCTAN FUNCTION AT BEGINNING OF
   [illegible]ROUTINE, BUT = ACCUMULATIVE ANGLE WHEN RETURNED
   IN ARGUMENT LIST.

   [illegible]JUSTMENT FOR ACCUMULATIVE ANGLE.

   [illegible]=-1.0 IF ANGLE IS DECREASING, =+1.0 IF ANGLE IS
```

```
C      INCREASING.
C
C      IRR=NUMBER OF RECORD OF DATA BEING PROCESSED IN
C      MAINLINE.
C
C      I=NUMBER OF DATA POINT BEING CONSIDERED IN RECORD OF
C      DATA BEING PROCESSED IN MAINLINE.
C
C      BITO=THE SIGN OF Q ON THE PREVIOUS EVALUATION
C      OF ANGLE.
C
C      STORE=VALUE OF ARCTAN FUNCTION ON PREVIOUS
C      EVALUATION OF THE ANGLE.
C
C      BIT1=THE SIGN OF'Q' ON THE PRESENT EVALUATION OF
C      ANGLE.
C
C
C*******************************************************
C
C

       SUBROUTINE ANGLE(P,Q,A,ADJ,BIT2,IRR,I,BITO,STORE)

C      CHECK OF 'Q'=0, IF SO NEGLECT POINT AND RETURN TO
C      MAINLINE.

       IF(ABS(Q)-0.0)24,21,24

C      EVALUATE ARCTAN TO OBTAIN UNADJUSTED ANGLE IN
C      DEGREES.

 24    A=ATAN(P/Q)*180.0/3.1416

C      CHECK FOR FIRST RECORD.

       IF(IRR-1) 2,2,5

C      CHECK FOR FIRST POINT IN FIRST RECORD.

 2     IF(I-1)4,4,5

C      SET SIGN OF 'Q' FOR FIRST ANGLE.

 4     BITO=Q/ABS(Q)

C      STORE VALUE OF FIRST ANGLE.

       STORE=A
```

```
      SUBROUTINE 'ANGLE'
C     SET SIGN OF 'Q' FOR PRESENT ANGLE.
 5    BIT1=Q/ABS(Q)
C     CHECK FOR A CHANGE OF SIGN IN 'Q'.
      IF(BIT1/BIT0)7,8,8
C     DETERMINE THE DIRECTION OF CHANGE FOR ANGLE.
 7    IF(STORE-5.0)13,14,14
C     ANGLE DECREASING.
 13   BIT2=-1.0
      GOTO 15
C     ANGLE INCREASING.
 14   BIT2=1.0
C     MODIFY ADJUSTMENT TO TAKE INTO ACCOUNT THE CHANGE OF
C     SIGN IN 'Q'.
 15   ADJ=ADJ+BIT2*180.0
C     SET SIGN OF 'Q' FOR NEXT TIME.
 8    BIT0=BIT1
C     STORE VALUE OF ANGLE FOR DIRECTION OF CHANGE CHECK ON
C     NEXT ITERATION.
      STORE=A
C     CALCULATE THE ADJUSTED ANGLE.
      A=A+ADJ
 21   RETURN
      END
```

PROGRAM 'BFKTY'

PROGRAM 'BFKTY'

```
C-----------------------------------------------------------------------
C                ** PROGRAM BFKTY **
C-----------------------------------------------------------------------
C
C        PURPOSE
C
C           TO UTILIZE THE REAL AND IMAGINARY COMPONENTS
C           OF THE FAST FOURIER TRANSFORM OF THE INPUT AND
C           OUTPUT PULSES TO OBTAIN THE AMPLITUDE RATIO,
C           PHASE LAG AND FREQUENCY CONTENT FUNCTION FOR
C           THE SYSTEM TESTED.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           IFLRI - FILE NO. FOR REAL PART OF TRANSFORM
C                   OF INPUT PULSE.
C           IFLII - FILE NO. FOR IMAG. PART OF TRANSFORM
C                   OF INPUT PULSE.
C           IFLRO - FILE NO. FOR REAL PART OF TRANSFORM
C                   OF OUTPUT PULSE.
C           IFLIO - FILE NO. FOR IMAG. PART OF TRANSFORM
C                   OF OUTPUT PULSE.
C           N     - NO. OF FREQUENCY POINTS USED.
C           AREA  - AREA UNDER INPUT PULSE.
C           RATE  - PULSE DATA SAMPLE RATE IN POINTS/SEC.
C           IREIM - =1, AMP RATIO, PHASE LAG, FCNT
C                   CALCULATED.
C                   =2, REAL AND IMAG. COMPONENTS OF
C                   TRANSFER FUNCTION CALCULATED.
C           SPECD - =0, FULL FREQUENCY CONTENT CURVE
C                   CALCULATED.
C                   =1, NORMALIZED FREQUENCY CONTENT
C                   CURVE CALCULATED.
C           POWR  - =0.5,FREQUENCY CONTENT CURVE
C                   CALCULATED.
C                   =1, SPECD=0, POWER SPECTRUM
C                   CALCULATED IN PLACE OF FREQUENCY
C                   CONTENT CURVE.
C
C
C        RESTRICTIONS
C
C           IF 'IREIM'=2, THE AMP RATIO, PHASE LAG AND
C           FREQ. CONTENT CAN ONLY BE OBTAINED IF THE
C           ORIGINAL TIME DATA IS TRANSFORMED AGAIN AND
C           'BFKTY' IS EXECUTED WITH 'IREIM'=1.
C
```

```
      PROGRAM 'BFKTY'

C
C              ** GENERAL COMMENTS **
C-----------------------------------------------------------------
C
C      THE DIMENSIONED VARIABLES ARI, AII, ARO, AIO
C      INITIALLY REPRESENT THE REAL AND IMAGINARY PARTS OF
C      THE FOURIER TRANSFORM OF THE INPUT AND OUTPUT PULSES
C      RESPECTIVELY.  AFTER A RECORD OF EACH VARIABLE IS
C      READ FROM DISK INTO CORE, THE VALUES IN EACH LOCATION
C      ARE OVERLAID WITH THE VALUES OF THE AMPLITUDE RATIO,
C      PHASE LAG AND FREQUENCY CONTENT (IREIM=2, REAL AND
C      IMAGINARY PARTS OF TRANSFER FUNCTION).  THESE VALUES
C      ARE THEN WRITTEN ON TO DISK IN THE SAME LOCATION THAT
C      ARI, AII, ARO AND AIO WERE READ FROM.
C
C
C          ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C      VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
C      NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C      OF THE APPROPRIATE SUBROUTINE.
C
C      LUR=LOGICAL UNIT NUMBER FOR KEYBOARD INPUT.
C
C      LUW=LOGICAL UNIT NUMBER FOR KEYBOARD OUTPUT.
C
C      NPPR=NUMBER OF DATA POINTS PER RECORD IN DISK FILES.
C
C      NREC=NUMBER OF FREQUENCY DATA POINTS USED.
C
C      DT=THE TIME INCREMENT BETWEEN TIME DATA POINTS.
C
C      L IS A TOTAL SUBSCRIPT FOR ALL THE FREQUENCY DATA
C      IN THE DISK FILES, IT IS USED IN THE GENERATION OF A
C      FILE OF FREQUENCY VALUES.
C
C      IRR=THE RECORD NUMBER ON DISK FILE READ WRITE
C      OPERATIONS.
C
C      DEN=DENOMINATOR TERM USED IN CALCULATING REAL AND
C      IMAGINARY PARTS OF TRANSFER FUNCTION.
C
C      GIM=IMAGINARY PART OF TRANSFER FUNCTION.
C
C      GRE=REAL PART OF TRANSFER FUNCTION.
C
C      A=PHASE ANGLE OF TRANSFER FUNCTION.
C
C      FCNT=FREQUENCY CONTENT OR SPECTRAL POWER OF INPUT
```

```
      PROGRAM 'BFKTY'
C     PULSE.
C
C
C-------------------------------------------------------------------
C
C
      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION ARI(40),AII(40),ARO(40),AIO(40)
C     INITIALIZE SOME PARAMETERS.
      LUR=1
      LUW=2
      NPPR=40
C     WRITE OUT PROGRAM IDENTIFICATION.
      WRITE(LUW,5555)
 5555 FORMAT('THIS IS AMP. RATIO, PHS. LAG, FREQ. CONTENT ',
     1'ROUTINE')
C     WRITE OUT DATA ENTRY INSTRUCTIONS.
      WRITE(LUW,3)
 3    FORMAT('ENTER,IFLRI,IFLII,IFLRO,IFLIO,N,AREA,RATE,',
     1'IREIM,SPECD,POWR')
C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.
      CALL FFINP(LUR,10,0,IFLRI,0,IFLII,0,IFLRO,0,IFLIO,0,N,
     11,AREA,1,RATE,0,IREIM,1,SPECD,1,POWR,IEROR)
      NREC=N/NPPR+1
      DT=1.0/RATE
      L=0
C     INITIALIZE PARAMETERS TO CALCULATE THE PHASE ANGLE.
      ADJ=0.0
      STORE=1.0
      BIT2=1.0
      BIT0=1.0
C     READ THE REAL AND IMAGINARY COMPONENTS OF THE INPUT
C     AND THE OUTPUT AND CALCULATE THE AMPLITUDE RATIO AND
C     PHASE ANGLE OF THE TRANSFER FUNCTION.
```

```
      PROGRAM 'BFKTY'

      DO 15 IRR=1,NREC

C     READ ONE RECORD OF DATA FROM DISK.

      READ(IFLRI'IRR)ARI
      READ(IFLII'IRR)AII
      READ(IFLRO'IRR)ARO
      READ(IFLIO'IRR)AIO

C     PROCESS EACH POINT IN EACH RECORD READ FROM DISK.

      DO 11 I=1,NPPR
      DEN=ARI(I)*ARI(I)+AII(I)*AII(I)
      GIM=(ARO(I)*AII(I)-AIO(I)*ARI(I))/DEN
      GRE=(ARI(I)*ARO(I)+AII(I)*AIO(I))/DEN

C     CHECK IF REAL AND IMAGINARY COMPONENTS OF TRANSFER
C     FUNCTION ARE ALL THAT IS REQUIRED.

      GOTO(30,31),IREIM
 31   ARI(I)=GRE
      AII(I)=GIM
      GOTO 11

C     CALL SUBROUTINE TO EVALUATE THE PHASE ANGLE.

 30   CALL ANGLE(GIM,GRE,A,ADJ,BIT2,IRR,I,BITO,STORE)

C     CALCULATE THE FREQUENCY CONTENT OR SPECTRAL POWER OF
C     INPUT PULSE.

      FCNT=((ARI(I)**2.+AII(I)**2.)/N/DT)**POWR
     1/(AREA**SPECD)

C     CALCULATE AMPLITUDE RATIO.

      ARI(I)=ABS((GRE**2.+GIM**2.)**0.5)
      AII(I)=A
      ARO(I)=FCNT

C     CALCULATE FREQUENCY VALUES.

      L=L+1
      AIO(I)=2*L*3.1416/N/DT
 11   CONTINUE

C     WRITE A RECORD OF AMP. RATIO, PHASE ANGLE, FREQUENCY
C     CONTENT OR SPECTRAL POWER, AND FREQUENCY ON DISK.

      WRITE(IFLRI'IRR)ARI
```

```
      PROGRAM 'BFKTY'

      WRITE(IFLII'IRR)AII
      WRITE(IFLRO'IRR)ARO
      WRITE(IFLIO'IRR)AIO
 15   CONTINUE

C     CHECK IF REAL AND IMAGINARY COMPONENTS OF TRANSFER
C     FUNCTION ARE ALL THAT IS REQUIRED.

      GOTO(32,33),IREIM

C     WRITE OUT NUMBERS OF FILES CONTAINING REAL AND
C     IMAGINARY COMPONENTS.

 33   WRITE(LUW,34)IFLRI,IFLII
 34   FORMAT('GRE IN FILE',I3,1X,'GIM IN FILE',I3)
      GOTO 53

C     WRITE OUT THE DISK FILE NOS. WHERE DATA IS STORED.

 32   WRITE(LUW,45)IFLRI
 45   FORMAT('AMP. RATIO IN FILE',I5)
      WRITE(LUW,46)IFLII
 46   FORMAT('PHASE LAG IN FILE',I5)
      WRITE(LUW,47)IFLRO
 47   FORMAT('FREQ. CONT. SMTH. IN FILE',I5)
      WRITE(LUW,48)IFLIO
 48   FORMAT('FREQUENCY IN FILE',I5)
 53   CALL LINK(CHIEF)
      END
```

PROGRAM 'BUCWS'

PROGRAM 'BUCWS'

```
C-----------------------------------------------------------------
C                ** PROGRAM BUCWS **
C-----------------------------------------------------------------
C
C        PURPOSE
C
C           TO PROVIDE FACILITY TO CALL SUBROUTINE 'ARPHF'
C           IN THE EVENT THAT 'ISMTH'=2 WHEN PROGRAM 'FREQ'
C           IS EXECUTED.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           THE PARAMETERS ENTERED HAVE THE SAME MEANING
C           AS THOSE GIVEN IN THE LISTING OF PROGRAM 'FREQ'.
C
C
C-----------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG
      LUR=1
      LUW=2
      NPPR=40
      DT=1.0
      RATE=1.0
      IRASS=1
      IFLAG=1

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,10)
 10   FORMAT('THIS IS PROGRAM BUCWS')

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

      WRITE(LUW,11)
 11   FORMAT('ENTER,IFLA,IFLB,IFLC,IFLD,NRF,AREA,FINT,FINC,'
     1,'SPECD,POWR')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,10,0,IFLA,0,IFLB,0,IFLC,0,IFLD,0,NRF,
```

PROGRAM 'BUCWS'

```
     11,AREA,1,FINT,1,FINC,1,SPECD,1,POWR,IEROR)
C     CALL SUBROUTINE TO CALCULATE THE AMPLITUDE RATIO,
C     PHASE ANGLE, FREQUENCY CONTENT AND FREQUENCY FOR THE
C     SYSTEM PULSE TESTED.
      CALL ARPHF(IFLA,IFLB,IFLC,IFLD,NRF,AREA,FINT,FINC,
     1SPECD,POWR)
      CALL LINK(CHIEF)
      END
```

PROGRAM 'PWRSL'

```
PROGRAM 'PWRSL'

C-----------------------------------------------------------------
C                 ** PROGRAM PWRSL **
C-----------------------------------------------------------------
C
C         PURPOSE
C            TO UTILIZE THE GENERAL PURPOSE PLOTTING
C            SUBROUTINE 'NWPLT'.
C
C
C         PARAMETERS ENTERED VIA KEYBOARD
C
C            NPTS,ID,IS,IG,IL,IT,IB,IU,IP,NFILX AND NFILY
C            ARE DEFINED IN THE LISTING OF SUBROUTINE
C            'NWPLT'.
C
C            IFLAG - =1, RETURN FOR  ANOTHER PLOT AFTER
C                    COMPLETION OF PRESENT PLOT.
C                    =2, EXIT AFTER COMPLETION OF PRESENT
C                    PLOT.
C
C
C            ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C      VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
C      NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C      OF THE APPROPRIATE SUBROUTINE.
C
C      LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C      LUW=LOGICAL UNIT NO. FOR TYPEWRITER OUTPUT.
C
C      NPPR=NO. OF POINTS PER RECORD.
C
C      VMIN IS A DUMMY VARIABLE USED IN CALCULATIONS TO
C      DETERMINE THE NO. OF CYCLES FOR MANUAL LOG SCALING.
C
C
C-----------------------------------------------------------------
C
C

       DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
      153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
      156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
       DIMENSION X(40),Y(40),LABX(10),LABY(10)

C      INTIIALIZE PARAMETERS.
```

```
          PROGRAM 'PWRSL'

      LUR=1
      LUW=2
      NPPR=40
      IR=2

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,5555)
 5555 FORMAT('THIS IS PLOTTING ROUTINE')

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

 50   WRITE(LUW,1)
 1    FORMAT('ENTER,NPTS,ID,IS,IG,IL,IT,IR,IU,IP,NFILX
     1,NFILY'
     1,',IFLAG')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,12,0,NPTS,0,ID,0,IS,0,IG,0,IL,0,IT,0,
     1IR,0,IU,0,IP,0,NFILX,0,NFILY,0,IFLAG,IEROR)

C     DETERMINE IF A FILE OF 'X' DATA IS TO BE GENERATED.

      IF(IT-1)13,13,14

C     DETERMINE IF MANUAL SCALING IS REQUIRED.

 13   IF(IS-1)23,23,14

C     WRITE OUT X-AXIS SCALING INSTRUCTIONS.

 14   WRITE(LUW,200)
 200  FORMAT('ENTER,XMIN,XMAX')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,2,1,XMIN,1,XMAX,IEROR)

C     DETERMINE IF X-AXIS IS A LOG SCALE.

      IF(IL-3)27,22,22

C     CALCULATE THE NO. OF LOG CYCLES ON X-AXIS.

 22   LC=0
      VMIN=XMIN
      VMAX=XMAX-0.01*XMAX
```

PROGRAM 'PWRSL'

```
 31    LC=LC+1
       VMIN=VMIN*10.0
       IF(VMAX-VMIN)32,32,31
 32    NCYCX=LC

C      DETERMINE IF MANUAL SCALING IS REQUIRED.

 27    IF(IS-1)23,23,19

C      WRITE OUT Y-AXIS SCALING INSTRUCTIONS.

 19    WRITE(LUW,201)
 201   FORMAT('ENTER,YMIN,YMAX')

C      CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C      ENTRY.

       CALL FFINP(LUR,2,1,YMIN,1,YMAX,IEROR)

C      DETERMINE IF Y-AXIS IS A LOG SCALE.

       IF(IL-3)42,23,41
 42    IF(IL-2)23,41,23

C      CALCULATE THE NO. OF LOG CYCLES ON Y-AXIS.

 41    LC=0
       VMAX=YMAX-0.01*YMAX
       VMIN=YMIN
 51    LC=LC+1
       VMIN=VMIN*10.0
       IF(VMAX-VMIN)52,52,51
 52    NCYCY=LC

C      DETERMINE IF LABELS ARE REQUIRED ON AXES.

 23    IF(IB-1)16,16,17

C      WRITE OUT INSTRUCTIONS TO ENTER LABEL FOR X-AXIS.

 16    WRITE(LUW,202)
 202   FORMAT('ENTER LABEL FOR X-AXIS')

C      CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C      ENTRY.

       CALL FFINP(LUR,1,102,LABX,IEROR)

C      WRITE OUT INSTRUCTIONS TO ENTER LABEL FOR Y-AXIS.
```

PROGRAM 'PWRSL'

```
      WRITE(LUW,203)
 203  FORMAT('ENTER LABEL FOR Y-AXIS')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,1,102,LABY,IEROR)

C     CALL GENERAL PURPOSE PLOTTING SUBROUTINE TO PLOT DATA.

 17   CALL NWPLT(X,Y,NPPR,ID,IS,IG,IL,IT,IB,IU,IR,IP,XMAX,
     1XMIN,NCYCX,YMAX,YMIN,NCYCY,LABX,LABY,NFILX,NFILY,NPTS)

C     DETERMINE IF ANOTHER PLOT IS REQUIRED.

      IF(IFLAG-1)50,50,61
 61   CALL LINK (CHIEF)
      END
```

SUBROUTINE 'NWPLT'

SUBROUTINE 'NWPLT'

```
C*****************************************
C************************************
C              **SUBROUTINE NWPLT**
C************************************************************
C
C        PURPOSE
C
C           A GENERAL PURPOSE PLOTTING SUBROUTINE
C
C        PARAMETERS
C
C           X      - INDEPENDENT VECTOR FOR PLOTTING, EMPTY
C           IF IT=2.
C           Y      - DEPENDENT VECTOR FOR PLOTTING
C           N      - NUMBER OF DATA POINTS IN X AND Y OR THE
C           NUMBER OF
C                    POINTS PER RECORD IF IR=2.
C           ID     - DEVICE, =1 FOR SCOPE, =2 FOR PLOTTER.
C           IS     - SCALING, =1 AUTOMATIC, =2 GIVEN.
C           IG     - POSITION, =1 TOP, =2 BOTTOM.
C           IL     - GRAPH, =1 LINEAR, =2 Y LOG - X LINEAR,
C           =3 Y LINEAR -
C                    X LOG, =4 LOG - LOG.
C           IT     - INDEPENDENT X, =1 GIVEN, =2 GENERATED
C           LINEAR BETWEEN
C                    XMIN AND XMAX.
C           IB     - LABELS, =1 AS GIVEN, =2 NO LABELS.
C           IU     - PLOT STATUS, =1 NEW PLOT, =2 LAST PLOT
C           (TOP), =3
C                    LAST PLOT (BOTTOM).
C           IR     - DATA SOURCE, =1 ALL DATA IN X AND Y, =2
C           DATA TO BE
C                    READ FROM A DISK FILE ONE RECORD AT A
C                    TIME.
C           IP     - PLOT TYPE, -VE LINE PLOT, =0 TO 5 POINT
C           PLOT WITH
C                    THE CORRESPONDING TYPE OF POINT.
C           XMAX   - MAXIMUM X VALUE IF IS=2 OR IT=2.
C           XMIN   - MINIMUM X VALUE IF IS=2 OR IT=2.
C           NCYCX  - NUMBER OF LOG CYCLES IF IS=2 AND ARE
C           REQUIRED
C           YMAX   - MAXIMUM Y VALUE IF IS=2.
C           YMIN   - MINIMUM Y VALUE IF IS=2.
C           NCYCY  - NUMBER OF LOG CYCLES IF IS=2 AND ARE
C           REQUIRED
C           LABX   - LABEL FOR X AXIS IN A2 FORMAT.
C           LABY   - LABEL FOR Y AXIS IN A2 FORMAT.
C                    NOTE...LABELS TO BE A MAXIMUM OF 20
```

```
      SUBROUTINE 'NWPLT'

C                    CHARACHTERS AND
C                    SHOULD BE DIMENSIONED 10.
C          NFILX - DATA FILE FOR X DATA IF IR=2.
C          NFILY - DATA FILE FOR Y DATA IF IR=2.
C          NPTS  - NUMBER OF POINTS  OF DATA TO BE TAKEN
C          FROM FILE NFIL
C
C       METHOD
C
C          WRITTEN BY R.B. NEWELL,DECEMBER 1968, U OF A.
C
C
C************************************
C*********************************
C
C

      SUBROUTINE NWPLT (X,Y,N,ID,IS,IG,IL,IT,IB,IU,IR,IP
     1,XMAX,XMIN,NCYC
     1 ,YMAX,YMIN,NCYCY,LABX,LABY,NFILX,NFILY,NPTS)
      DIMENSION X(1),Y(1),LABX(10),LABY(10)

C     FUNCTION FOR LOGARITHMS TO THE BASE 10.

      ALOG1(Z)=ALOG(Z)/2.302585

C     SECTION SETTING UP A SCOPE PLOT FOR ID=1.

      GO TO (1,4),ID
    1 CALL PLOTD(1)

C     ERASE SCOPE IF IT IS A NEW PLOT

      IF(IU+IG-2)3,2,3
    2 CALL ERASE
    3 CALL ORGSC(0.0,0.0)
      GO TO 5

C     SET UP THE PLOT ON THE PLOTTER IF ID=2.

    4 CALL PLOTD(0)

C     SECTION FOR SCALING THE VARIABLES. IF IS=1.

    5 GO TO (8,8,9,10),IL

C     LINEAR SCALING OF X.

    8 NX=10
      ITT=IT*IS
```

SUBROUTINE 'NWPLT'

```
      GO TO (108,107,107,107),ITT
  107 SCFX=(XMAX-XMIN)/NX
      GO TO 109
  108 CALL NWLNS (X,N,XMIN,XMAX,NX,SCFX,IR,NFILX,NPTS)
  109 GO TO (9,12,9,10),IL

C     LINEAR SCALING FOR Y.

    9 NY=5
      GO TO (115,116),IS
  116 SCFY=(YMAX-YMIN)/NY
      GO TO 117
  115 CALL NWLNS (Y,N,YMIN,YMAX,NY,SCFY,IR,NFILY,NPTS)
  117 GO TO (14,12,10,10),IL

C     LOGARITHMIC SCALING OF X.

   10 GO TO (80,110),IS
   80 CALL NWLGS (X,N,XMIN,XMAX,NCYCX,IR,NFILX,NPTS)
      IF(NCYCX)100,100,110

C     CONVERT X VALUES TO LOGARITHMS FOR PLOTTING.

  110 XMIN=ALOG1(XMIN)
      XMAX=ALOG1(XMAX)
      DO 11 I=1,N
      X(I)=ALOG1(X(I))
   11 CONTINUE
      GO TO (14,12,14,12),IL

C     LOGARITHMIC SCALING OF Y.

   12 GO TO (81,112),IS
   81 CALL NWLGS (Y,N,YMIN,YMAX,NCYCY,IR,NFILY,NPTS)
      IF(NCYCY)100,100,112

C     CONVERT Y VALUES TO LOGARITHMS FOR PLOTTING.

  112 YMIN=ALOG1(YMIN)
      YMAX=ALOG1(YMAX)
      DO 13 I=1,N
      Y(I)=ALOG1(Y(I))
   13 CONTINUE

C     CALCULATING THE SCALE FACTORS.

   14 XIPUU=7.5/(XMAX-XMIN)
      YIPUU=4.5/(YMAX-YMIN)

C     LOCATING THE PRESENT PEN POSITION
```

SUBROUTINE 'NWPLT'

```
      GO TO (15,16,17),IU
   15 GO TO (16,17),IG
   16 YPOS=YMIN-5.6/YIPUU
      GO TO 18
   17 YPOS=YMIN-0.5/YIPUU
   18 GO TO (19,20),ID
   19 XPOS=XMIN-0.5/XIPUU
      GO TO 21
   20 IF(IU-2)119,121,121
  119 GO TO (19,121),IG
  121 XPOS=XMIN+8.5/XIPUU

C     CALL SCALING SUBROUTINE

   21 CALL SCALF (XIPUU,YIPUU,XPOS,YPOS)
      GO TO (122,50,50),IU

C     MARKING THE GRID LINES ON THE PLOT.

  122 IF(YMAX)24,23,22
   22 IF(YMIN)25,24,24
   23 POS=YMAX
      GO TO 26
   24 POS=YMIN
      GO TO 26
   25 POS=0.0
   26 GO TO (27,27,28,28),IL

C     MARK A LINEAR X GRID.

   27 CALL FGRID (0,XMIN,POS,SCFX,NX)
      GO TO 29

C     MARK A LOGARITHMIC X GRID

   28 CSIZE=(XMAX-XMIN)/NCYCX
      CALL LGRID(0,XMIN,POS,XIPUU,YIPUU,0,CSIZE,NCYCX)
   29 IF(XMAX)32,31,30
   30 IF(XMIN)33,32,32
   31 POS=XMAX
      GO TO 34
   32 POS=XMIN
      GO TO 34
   33 POS=0.0
   34 GO TO (35,36,35,36),IL

C     MARK A LINEAR Y GRID.

   35 CALL FGRID (1,POS,YMIN,SCFY,NY)
      GO TO 37
```

[illegible]

```
      SUBROUTINE 'NWPLT'

C     MARK A LOGARITHMIC Y GRID.

   36 CSIZE=(YMAX-YMIN)/NCYCY
      CALL LGRID (1,POS,YMIN,XIPUU,YIPUU,0,CSIZE,NCYCY)
   37 CONTINUE

C     ANNOTATION OF THE GRIDS
C

      GO TO (40,40,41,41),IL

C     ON LINEAR X GRID

   40 CALL NWANN (0,XMIN,YMIN,XIPUU,YIPUU,XMIN,XMAX,SCFX,IB
     1,LABX)
      GO TO (42,43,50,50),IL

C     ON LOG X GRID

   41 I=IFIX(XMIN+0.05*XMIN/ABS(XMIN))
      AMIN=10.0**I
      I=IFIX(XMAX+0.05*XMAX/ABS(XMAX))
      AMAX=10.0**I
      CALL NWANN (0,XMIN,YMIN,XIPUU,YIPUU,AMIN,AMAX,0.0,IB
     1,LABX)
      GO TO (50,50,42,43),IL

C     ON THE LINEAR Y GRID

   42 CALL NWANN (1,XMIN,YMIN,XIPUU,YIPUU,YMIN,YMAX,SCFY,IB
     1,LABY)
      GO TO 50

C     ON THE LOG Y GRID

   43 I=IFIX(YMIN+0.05*YMIN/ABS(YMIN))
      AMIN=10.0**I
      I=IFIX(YMAX+0.05*YMAX/ABS(YMAX))
      AMAX=10.0**I
      CALL NWANN (1,XMIN,YMIN,XIPUU,YIPUU,AMIN,AMAX,0.0,IB
     1,LABY)

C     PLOTTING THE POINTS
C     GENERATE A X VALUE IF IT=2

   50 NN=N
      GO TO (165,153),IR

C     LOCATING THE FIRST PEN POSITION IF AN ARRAY PLOT
```

SUBROUTINE 'NWPLT'

```
  165 GO TO (60,61),IT
   61 X(1)=XMIN
   60 CALL FPLOT (1,X(1),Y(1))
      IF(IP)51,152,152
   51 CALL FPLOT (2,X(1),Y(1))
C     SETTING LOOP VARIABLES IF AN ARRAY PLOT
  152 NUM=1
      NPTS=N
      GO TO 154
C     SETTING LOOP VARIABLES IF A FILR PLOT
  153 NUM=1+(NPTS-N)/N
      IF(NPTS-N)86,85,85
   85 NN=N
      GO TO 154
   86 NN=NPTS
C     OUTSIDE LOOP FOR FILE RECORDS
  154 DO 55 J=1,NUM
C     OBTAINING DATA FROM FILE
      GO TO (156,155),IR
C     READ JUST THE Y FILE IF X TO BE GENERATED
  155 GO TO (157,158),IT
C     READ X DATA
  157 READ(NFILX'J)(X(IJ),IJ=1,N)
C     LOGS IF REQUIRED
      GO TO (158,158,70,70),IL
   70 DO 71 I=1,N
      X(I)=ALOG1(X(I))
   71 CONTINUE
C     READ Y DATA
  158 READ(NFILY'J)(Y(IJ),IJ=1,N)
C     LOGS IF REQUIRED
      GO TO (156,72,156,72),IL
```

```
     SUBROUTINE 'NWPLT'

 72 DO 73 I=1,N
    Y(I)=ALOG1(Y(I))
 73 CONTINUE


C   INNER LOOP FOR GRAPH PLOTTING

156 DO 55 I=1,NN
    IF(I+J-2)167,168,167
168 IJ=-2
    GO TO 169
167 IJ=0
169 GO TO (62,63),IT

C   GENERATE A X VALUE IF IT=2

 63 X(I)=XMIN+(XMAX-XMIN)*(N*(J-1)+I-1)/(NPTS-1)
 62 IF(IP)53,54,54

C   LINE PLOT - PEN IS DOWN.

 53 CALL FPLOT (IJ,X(I),Y(I))
    GO TO 55

C   POINT PLOT - PEN IS NOW UP.

 54 CALL FPLOT (-2,X(I),Y(I))
    CALL POINT (IP)
    CALL FPLOT (1,X(I),Y(I))
 55 CONTINUE

C   DETERMINE IF FINISHED

    NN=NPTS-N*NUM
    IF(NN)160,160,162
162 NUM=NUM+1
    J=NUM
    GO TO 155
160 GO TO (57,56),ID
 56 XPOS=XMIN+8.5/XIPUU
 57 CALL FPLOT (1,XPOS,YPOS)
100 RETURN
    END
```

SUBROUTINE 'NWLNS'

SUBROUTINE 'NWLNS'

```
C****************************************
C*********************************
C                **SUBROUTINE NWLNS**
C**********************************************************************
C
C        PURPOSE
C
C           SUBROUTINE TO DETERMINE A SUITABLE LINEAR SCALE
C           FOR A GIVEN
C           VECTOR OF VALUES
C
C        PARAMETERS
C
C           X      - VECTOR OF DATA TO BE SCALED.
C           N      - NUMBER OF POINTS IN THE VECTOR X.
C           NMIN   - CALCULATED MINIMUM VALUE FOR THE SCALE
C           NMAX   - CALCULATED MAXIMUM VALUE FOR THE SCALE
C           NX     - APPROXIMATE NUMBER OF INTERVALS WANTED
C           ON THE AXIS
C                    REQUIRED AND THE CALCULATED NUMBER
C                    RETURNED.
C           SS     - LENGTH OF THE SCALE INTERVAL IN USERS
C           UNITS.
C           IR     - DATA SOURCE INDICATOR, =1 ALL DATA IN X,
C           =2 DATA IN
C                    A DISK DATA FILE.
C           NFIL   - IDENTIFICATION NUMBER OF THE DATA FILE.
C           NREC   - NUMBER OF RECORDS OF DATA IN THE FILE.
C
C        METHOD
C
C           WRITTEN BY R.B. NEWELL,DECEMBER 1968, U OF A.
C
C
C****************************************
C*********************************
C
C


      SUBROUTINE NWLNS (X,N,NMIN,NMAX,NX,SS,IR,NFIL,NREC)
      REAL NMAX,MAX,NMIN,MIN,X(1)

C     DETERMINATION OF THE TRUE MAX AND MIN OF THE GIVEN
C     DATA.

      CALL NWMAX (X,N,MIN,MAX,IR,0,NFIL,NREC)
      RGE=(MAX-MIN)/NX
      IF(RGE-1.0)1,2,2
```

SUBROUTINE 'NWLNS'

```
 1 IY=-1
   GO TO 3
 2 IY=0
 3 IX=IFIX(ALOG(RGE)/2.302585)+IY
   RDRGE=RGE/10.0**IX
   IF(RDRGE-1.0)14,13,14
13 RDRGE=10.0
   IX=IX-1
   GO TO 15
14 DO 4 I=3,20
   IF(RDRGE-0.5*I)5,5,4
 4 CONTINUE
   RDRGE=10.0
   GO TO 15
 5 RDRGE=0.5*I
15 SS=RDRGE*10.0**IX
   IF(MIN)16,17,17
16 IY=-1
   GO TO 18
17 IY=0
18 NMIN=SS*IFIX(MIN/SS)+IY*SS
   NMAX=NMIN+NX*SS
19 IF(NMAX-MAX)6,7,7
 6 NMAX=NMAX+SS
   NX=NX+1
   GO TO 19
 7 RETURN
   END
```

SUBROUTINE 'NWMAX'

SUBROUTINE 'NWMAX'

```
C*****************************************
C************************************
C                **SUBROUTINE NWMAX**
C************************************************************
C
C        PURPOSE
C
C           SUBROUTINE TO FIND THE MIN AND MAX OF A SERIES
C           OF DATA AND
C           TO INDICATE ZERO OR NEGATIVE NUMBERS IF
C           REQUESTED.
C
C        PARAMETERS
C
C           X      - VECTOR OF DATA.
C           N      - NUMBER OF POINTS OF DATA IN X.
C           XMIN   - MINIMUM OF THE GIVEN DATA.
C           XMAX   - MAXIMUM OF THE GIVEN DATA.
C           IR     - DATA INDICATOR, =1 ALL DATA IN X, =2
C           DATA ON DISK.
C           II     - REQUEST AND ERROR INDICATOR, +VE IF
C           ZERO/NEGATIVE
C                    CHECK REQUIRED, =0 NO CHECK, -VE IF
C                    ZERO/NEGATIVE
C                    DATA FOUND.
C           NFIL   - DISK FILE IDENTIFICATION NUMBER.
C           NPTS   - NUMBER OF POINTS  OF DATA, 'N' POINTS
C           PER RECORD.
C
C        METHOD
C
C           WRITTEN BY R.B. NEWELL,DECEMBER 1968, U OF A.
C
C*****************************************
C************************************
C

      SUBROUTINE NWMAX (X,N,XMIN,XMAX,IR,II,NFIL,NPTS)
      DIMENSION X(1)
      XMAX=-1.E30
      XMIN=1.E30
      NN=N
      GO TO (1,2),IR
    1 NUM=1
      NPTS=N
      GO TO 3
    2 NUM=NPTS/N
      IF(NUM)21,21,3
```

SUBROUTINE 'NWMAX'

```
 21 NN=NPTS
  3 DO 12 I=1,NUM
    GO TO (5,4),IR
  4 READ(NFIL'I)(X(IJ),IJ=1,NN)
  5 DO 12 J=1,NN
    IF(X(J)-XMIN)6,7,7
  6 XMIN=X(J)
  7 IF(X(J)-XMAX)9,9,8
  8 XMAX=X(J)
  9 IF(II)12,12,10
 10 IF(X(J))11,11,12
 11 WRITE(2,101)
101 FORMAT(' ERROR - ZERO OR NEG DATA GIVEN TO LOG
   1 SCALING')
    II=-1
    RETURN
 12 CONTINUE
    NN=NPTS-N*NUM
    IF(NN)14,14,13
 13 NUM=NUM+1
    I=NUM
    GO TO 4
 14 RETURN
    END
```

SUBROUTINE 'NWLGS'

SUBROUTINE 'NWLGS'

```
C*************************************
C*************************************
C                **SUBROUTINE NWLGS**
C*************************************************************
C
C        PURPOSE
C
C           SUBROUTINE TO DETERMINE A LOGARITHMIC SCALE FOR
C           A GIVEN SET
C           OF VALUES TO AN INTEGER NUMBER OF CYCLES.
C
C        PARAMETERS
C
C           X      - VECTOR OF GIVEN VALUES
C           N      - NUMBER OF VALUES IN X.
C           XMIN   - MINIMUM VALUE OF THE RESULTING SCALE.
C           XMAX   - MAXIMUM VALUE OF THE RESULTING SCALE.
C           NC     - NUMBER OF DETERMINED CYCLES FOR THE
C           SCALE.
C           IR     - DATA INDICATOR, =1 ALL DATA IN X, =2
C           DATA ON DISK
C                    FILE, N= NUMBER OF POINTS PER RECORD.
C           NFIL   - DISK FILE IDENTIFICATION NUMBER.
C           NREC   - NUMBER OF RECORDS IN THE FILE.
C
C        METHOD
C
C           WRITTEN BY R.B. NEWELL,DECEMBER 1968, U OF A.
C
C
C
C*************************************
C*************************************
C
C
      SUBROUTINE NWLGS (X,N,XMIN,XMAX,NC,IR,NFIL,NREC)
      DIMENSION X(1)
C     FUNCTION FOR LOGARITHMS TO THE BASE 10.
      ALOG1(Z)=ALOG(Z)/2.302585
      IX=1
      II=1
C     CALL SUBROUTINE FOR DATA MAX AND MIN AND A ZERO/SIGN
C     CHECK.
```

SUBROUTINE 'NWLGS'

```
      CALL NWMAX (X,N,XMIN,XMAX,IR,II,NFIL,NREC)
C     RETURN IF A ZERO OR NEGATIVE VALUE FOUND.
      IF(II)1,2,2
    1 NC=-1
      RETURN
    2 IF(XMAX-1.0)14,3,4
    3 NC=0
      GO TO 5
   14 IX=0
    4 NC=IFIX(ALOG1(XMAX))+IX
    5 XMAX=10.0**NC
      IX=1
      IF(XMIN-1.0)7,6,17
    6 INC=0
      GO TO 8
   17 IX=0
    7 INC=IFIX(ALOG1(XMIN))-IX
    8 XMIN=10.0**INC
      NC=NC-INC
      RETURN
      END
```

SUBROUTINE 'LGRID'

SUBROUTINE 'LGRID'

```
C************************************************************
C                  **SUBROUTINE LGRID**
C************************************************************
C
C      PURPOSE
C
C          THIS GRID SUBROUTINE PLOTS A STRAIGHT LINE
C          PARALLEL TO EITHER
C          THE X OR Y AXIS IN A POSITIVE OR NEGATIVE
C          DIRECTION . IT
C          PLACES TICK MARKS ON THIS LINE TO GIVE A
C          LOGARITHMIC CYCLE .
C
C
C      USAGE
C
C          CALL LGRID(IND,XPS,YPS,XSCAL,YSCAL,IPOS,CSIZE
C          ,NCYLE)
C
C
C      DESCRIPTION OF PARAMETERS
C
C          IND - INTEGER CONSTANT THAT SPECIFIES THE
C          DIRECTION
C              OF THE GRID LINE .
C              IND=0 + VE X - DIRECTION
C              IND=1 +VE  Y - DIRECTION
C              IND=2 -VE X - DIRECTION
C              IND=3 -VE Y - DIRECTION
C          XPS , YPS - THE X AND Y VALUES , MEASURED IN USERS
C          UNITS ,
C          UPON EXITING THESE VARIABLES WILL CONTAIN THE
C          PRESENT PEN
C          POSITION IN USERS UNITS .
C          XSCAL - SCALE FOR THE X - DIRECTION , INCHES/USERS
C          UNIT .
C          YSCAL - SCALE FOR THE Y - DIRECTION , INCHES/USERS
C          UNIT .
C          IPOS - AN INTEGER CONSTANT THAT SPECIFIES WHETHER
C          XPS AND
C          YPS VALUES ARE STARTING OR ENDING POINTS .
C              IPOS=0 STARTING POINT
C              IPOS=1 ENDING POINT
C          CSIZE - A REAL CONSTANT OR VARIABLE THAT DEFINES
C          THE LENGTH
C              OF A CYCLE , MEASURED IN USERS UNITS .
C          NCYLE - NUMBER OF FULL LOGARITHMIC CYCLES TO BE
C          PLOTTED .
```

```
      SUBROUTINE 'LGRID'
C
C
C     SUBROUTINES AND FUNCTIONAL SUBPROGRAMS REQUIRED
C
C         FPLOT
C
C
C     METHOD
C
C         WRITTEN BY R.A.FARWELL , JULY 1968 , U. OF A.
C
C
C*******************************************************************
C
C
      SUBROUTINE LGRID (IND,XPOS,YPOS,XSCAL,YSCAL,IPOS,CSIZE
     1,NCYLE)
      DIMENSION PAR(9)
      DATA PAR/ 0.30103,0.17609,0.12494,0.09691,0.07918
     1,0.06695,0.05799
     1          0.05115,0.04576/
      XPS=XPOS
      YPS=YPOS
C     MOVE PEN TO STARTING POINT
      CALL FPLOT(+5,XPOS,YPOS)
      CSIZ=CSIZE
      IF(IND-1) 1,2,3
C     X-DIRECTION
 1    YP1=0.05/YSCAL
      XP1=0.0
      IX=1
      GO TO 4
C     Y - DIRECTION
 2    YP1=0.0
      XP1=0.05/XSCAL
      IX=0
      GO TO 4
 3    CSIZ=-CSIZ
      IF(IND-3) 1,2,2
 4    CALL FPLOT(6,XPOS+1.5*XP1,YPOS+1.5*YP1)
      CALL FPLOT(0,XPOS-1.5*XP1,YPOS-1.5*YP1)
      CALL FPLOT(0,XPOS,YPOS)
      CXSIZ=IX*CSIZ
```

SUBROUTINE 'LGRID'

```
      CYSIZ=(1-IX)*CSIZ
      DO 7 I=1,NCYLE
      DO 6 II=1,9
      IJ=10*IPOS+II*(1-2*IPOS)
      XPOS=XPOS+PAR(IJ)*CXSIZ
      YPOS=YPOS+PAR(IJ)*CYSIZ
      CALL FPLOT(0,XPOS,YPOS)
      IF(II-9) 5,6,6
5     CALL FPLOT(0,XPOS+XP1,YPOS+YP1)
      CALL FPLOT(0,XPOS-XP1,YPOS-YP1)
      CALL FPLOT(0,XPOS,YPOS)
6     CONTINUE
      CALL FPLOT(0,XPOS+1.5*XP1,YPOS+1.5*YP1)
      CALL FPLOT(0,XPOS-1.5*XP1,YPOS-1.5*YP1)
7     CALL FPLOT(0,XPOS,YPOS)
      XPOS=XPS
      YPOS=YPS
      RETURN
      END
```

SUBROUTINE 'NWANN'

SUBROUTINE 'NWANN'

```
C****************************************
C***************************************
C                **SUBROUTINE NWANN**
C***************************************************************
C
C        PURPOSE
C
C           A SUBROUTINE TO PLACE LABELS AND GRID SCALES ON
C           AN AXIS.
C
C        PARAMETERS
C
C           IAX    - AXIS INDICATOR, =0 X AXIS, =1 Y AXIS.
C           XPOS   - X COORDINATE OF THE LOWER LEFT CORNER.
C           YPOS   - Y COORDINATE OF THE LOWER LEFT CORNER.
C           XIPUU  - X AXIS SCALE IN INCHES PER USER'S UNIT.
C           YIPUU  - Y AXIS SCALE IN INCHES PER USER'S UNIT.
C           AMIN   - MINIMUM TO WRITE ON THE GRID.
C           AMAX   - MAXIMUM TO WRITE ON THE GRID.
C           SCF    - GRID INTERVAL FOR WRITING ON THE GRID.
C           IB     - LABEL INDICATOR, =1 LABEL AS GIVEN, =2
C           NO LABEL.
C           ALAB   - VECTOR CONTAINING THE LABEL TO A MAXIMUM
C           OF 20
C                    CHARACHTERS UNDER AN A2 FORMAT.
C
C        METHOD
C
C           WRITTEN BY R.B. NEWELL,DECEMBER 1968, U OF A.
C
C****************************************
C***************************************
C
C

      SUBROUTINE NWANN (IAX,XPOS,YPOS,XIPUU,YIPUU,AMIN,AMAX
     1,SCF,IB,ALAB
      INTEGER ALAB(10)
      THETA=IAX*1.57
      GO TO (1,3),IB
    1 X=(1-IAX)*(XPOS+1.5/XIPUU)+IAX*(XPOS-0.3/XIPUU)
      Y=(1-IAX)*(YPOS-0.5/YIPUU)+IAX*YPOS
      CALL FCHAR (X,Y,0.15,0.20,THETA)
      WRITE(7,2)(ALAB(LL),LL=1,10)
    2 FORMAT(10A2)
    3 AX=IAX*(XPOS-0.1/XIPUU)
      AY=(1-IAX)*(YPOS-0.25/YIPUU)
      X=(1-IAX)*XPOS+AX
```

SUBROUTINE 'NWANN'

```
      Y=AY+IAX*YPOS
      CALL FCHAR (X,Y,0.10,0.15,THETA)
      WRITE(7,4)AMIN
    4 FORMAT(E10.3)
      X=(1-IAX)*(XPOS+2.5/XIPUU)+AX
      Y=AY+IAX*(YPOS+1.0/YIPUU)
      CALL FCHAR (X,Y,0.10,0.15,THETA)
      IF(SCF)7,6,7
    6 WRITE(7,9)
    9 FORMAT(4X,'LOG  CYCLES')
      GO TO 8
    7 WRITE(7,5)SCF
    5 FORMAT(2X,'(',E10.3,' INTERVALS )')
    8 X=(1-IAX)*(XPOS+6.5/XIPUU)+AX
      Y=AY+IAX*(YPOS+3.5/YIPUU)
      CALL FCHAR (X,Y,0.10,0.15,THETA)
      WRITE(7,4)AMAX
      RETURN
      END
```

PROGRAM 'PICK'

PROGRAM 'PICK'

```
C-----------------------------------------------------------------
C                 ** PROGRAM PICK **
C-----------------------------------------------------------------
C
C        PURPOSE
C
C           TO UTILIZE THE SUBROUTINE 'REDPT' TO PERMIT
C           REDUCTION IN THE NUMBER OF DATA POINTS IN A
C           FILE.
C
C           TO GENERATE A FILE OF TIME VALUES USED IN
C           PLOTTING TIME DATA.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           IFLR, IFLW, NREC, NPICK ARE EXPLAINED IN
C                    SUBROUTINE 'REDPT'.
C           RATE  - SAMPLE RATE FOR UNREDUCED DATA IN
C                   POINTS PER SECOND.
C           ITIME - = 1 REDUCE DATA DO NOT GENERATE FILE
C                   OF TIME VALUES.
C                   = 2 REDUCE DATA, GENERATE FILE TIME
C                   VALUES.
C                   = 3 GENERATE A FILE OF TIME VALUES
C                   ONLY.
C           IFLT  - = NO. OF FILE FOR TIME VALUES.
C           IFLAG - = 1, RETURN FOR FURTHER ASSIGNMENT.
C                   = 2, EXIT.
C
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------
C
C     VARIABLES IN THE SUBROUTINE ARGUMENT LISTS WHICH ARE
C     NOT EXPLAINED IN THIS LISTING ARE COVERED IN THAT
C     OF THE SUBROUTINE.
C
C     LUR=LOGICAL UNIT NO. FOR TYPEWRITER INPUT.
C
C     LUW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT.
C
C     JFLAG=1 NO DATA REDUCTION TOOK PLACE, =2 DATA
C     REDUCTION OCCURRED.
C
C     NPPR=NUMBER OF DATA POINTS PER RECORD.
C
C
```

```
          PROGRAM 'PICK'
C      IFLAG=1 RETURN FOR ANOTHER RUN, =2 EXIT.
C
C      IW=RECORD NO. ON DISK WRITE OPERATION.
C
C
C-----------------------------------------------------------------
C
C
       DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
      153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
      156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
       DIMENSION V(40)
       COMMON LUR,LUW,NPPR,DT,RATE,IRASS,IFLAG
       LUR=1
       LUW=2
       JFLAG=1
       NPPR=40
C      WRITE OUT PROGRAM IDENTIFICATION.
       WRITE(LUW,60)
 60    FORMAT('THIS IS PROGRAM PICK')
C      WRITE OUT DATA ENTRY INSTRUCTIONS.
 10    WRITE(LUW,50)
 50    FORMAT('ENTER,IFLR,IFLW,NREC,NPICK,IRST,RATE,ITIME',
      1',IFLT,IFLAG')
C      CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C      ENTRY.
       CALL FFINP(LUR,9,0,IFLR,0,IFLW,0,NREC,0,NPICK,0,IRST,1
      1,RATE,0,ITIME,0,IFLT,0,IFLAG,IEROR)
       DT=1.0/RATE
C      CHECK IF A FILE OF TIME DATA IS REQUIRED OR NOT.
       GOTO(40,41,42),ITIME
 40    JFLAG=2
C      CALL SUBROUTINE TO REDUCE NUMBER OF DATA POINTS.
 41    CALL REDPT(IFLR,IFLW,NREC,IRW,NPICK,IRST)
C      CHECK IF DATA WAS REDUCED.
       IF(JFLAG-1)43,43,44
```

```
      PROGRAM 'PICK'
C     GENERATE A FILE OF TIME VALUES.
 43   NREC=IRW
 42   T=0.0
      DO 15 IW=1,NREC
      DO 16 J=1,NPPR
      V(J)=T
      T=T+DT
 16   CONTINUE
C     STORE A RECORD OF TIME VALUES.
      WRITE(IFLT'IW)V
 15   CONTINUE
 44   IF(IFLAG-1)10,10,20
 20   CALL LINK(CHIEF)
      END
```

PROGRAM 'WRITE'

PROGRAM 'WRITE'

```
C-----------------------------------------------------------------------
C              ** PROGRAM WRITE **
C-----------------------------------------------------------------------
C
C        PURPOSE
C
C           UTILITY PROGRAM TO MANIPULATE DATA FILES.
C           PROVIDES FACILITY TO READ DATA FROM A FILE AND
C           WRITE IT -
C             1. INTO ANOTHER FILE.
C             2. PUNCH IT ONTO CARDS IN 'E' FORMAT.
C             3. WRITE IT ONTO THE LINE PRINTER.
C
C           THE CAPABILITY TO HANDLE BOTH REAL AND INTEGER
C           VALUES IS AVAILABLE.
C
C
C        PARAMETERS ENTERED VIA KEYBOARD
C
C           IFLR  - NO. OF DISK FILE FROM WHICH DATA IS
C                   READ.
C           IFLW  - NO. OF DISK FILE WHERE DATA IS WRITTEN.
C           IORF  - = 2 FOR INTEGER, =2 FOR REAL.
C           IWRIT - = 1, WRITE IN DISK FILE.
C                   = 2, PUNCH ON CARDS.
C                   = 3, WRITE ON LINE PRINTER.
C           NREC  - = NO. OF RECORDS READ FROM IFLR.
C           IRETN - = 1 RETURN FOR ANOTHER RUN, = 2 EXIT.
C
C
C           ** TABLE OF FUNCTION OF VARIABLE NAMES **
C-----------------------------------------------------------------------
C
C      ADATA DIMENSIONED VARIABLE INTO WHICH EACH RECORD OF
C      REAL VALUES IS READ DURING DATA TRANSFER.
C
C      IDATA DIMENSIONED VARIABLE INTO WHICH EACH RECORD OF
C      INTEGER VALUES IS READ DURING DATA TRANSFER.
C
C      LUR=LOGICAL UNIT NUMBER FOR TYPEWRITER INPUT.
C
C      LUW=LOGICAL UNIT NUMBER FOR TYPEWRITER OUTPUT.
C
C      LUP=LOGICAL UNIT NUMBER FOR CARD PUNCHING.
C
C      LPR=LOGICAL UNIT NUMBER FOR LINE PRINTER OUTPUT.
C
C      NPPRI=NO. OF INTEGER VALUES PER RECORD.
```

```
      PROGRAM 'WRITE'
C
C     NPPRF=NO. OF REAL VALUES PER RECORD.
C
C     KK=NO. OF RECORD ON FILE READ OPERATION.
C
C
C-------------------------------------------------------------------
C
C

      DEFINE FILE 51(64,80,U,J0),52(64,80,U,J0),
     153(64,80,U,J0),54(64,80,U,J0),55(64,80,U,J0),
     156(64,80,U,J0),57(64,80,U,J0),58(64,80,U,J0)
      DIMENSION ADATA(40),IDATA(80)

C     INITIALIZE PARAMETERS.

      LUR=1
      LUW=2
      LUP=5
      LPR=6
      NPPRI=80

C     WRITE OUT PROGRAM IDENTIFICATION.

      WRITE(LUW,5555)
 5555 FORMAT('THIS IS PROGRAM WRITE')
      NPPRF=NPPRI/2

C     WRITE OUT DATA ENTRY INSTRUCTIONS.

 222  WRITE(LUW,3)
 3    FORMAT('ENTER,IFLR,IFLW,IORF,IWRIT,NREC,IRETN')

C     CALL SYSTEM SUBROUTINE TO GIVE FREE FORMAT ON DATA
C     ENTRY.

      CALL FFINP(LUR,6,0,IFLR,0,IFLW,0,IORF,0,IWRIT,0,NREC,0
     1,IRETN,
     1IEROR)
      DO 100 KK=1,NREC

C     DETERMINE WHETHER RECORD CONTAINS INTEGER OR REAL
C     VALUES.

      GOTO(14,15),IORF
 14   READ(IFLR'KK) IDATA
      GOTO 888
 15   READ(IFLR'KK) ADATA
```

```
      PROGRAM 'WRITE'
C     DETERMINE WHERE TO WRITE DATA.
 888  GOTO (16,17,18),IWRIT
C     DETERMINE WHETHER DATA WRITTEN ON DISK IS INTEGER OR
C     REAL.
 16   GOTO(19,20),IORF
 19   WRITE(IFLW'KK) IDATA
      GOTO 100
 20   WRITE(IFLW'KK) ADATA
      GOTO 100
C     READ A BLANK CARD.
 17   READ(5,4002)
 4002 FORMAT(' ')
C     DETERMINE WHETHER DATA PUNCHED IS INTEGER OR REAL.
      GOTO(21,22),IORF
 21   WRITE(LUP,50) IDATA
 50   FORMAT(10I6)
      GOTO 100
 22   WRITE(LUP,51) ADATA
 51   FORMAT(5F10.6)
      GOTO 100
C     DETERMINE WHETHER DATA WRITTEN ON LINE PRINTER IS
C     INTEGER OR REAL.
 18   GOTO(23,24),IORF
 23   WRITE (LPR,55) IDATA
 55   FORMAT(17(1X,I6))
      GOTO 100
 24   WRITE(LPR,56) ADATA
 56   FORMAT(9(1X,E12.5))
 100  CONTINUE
C     DETERMINE WHETHER MORE DATA IS TO BE WRITTEN.
      GOTO (222,444),IRETN
 444  CALL LINK(CHIEF)
      END
```

SUBROUTINE 'LINEA'

```
SUBROUTINE 'LINEA'

C*****************************************************************
C                **SUBROUTINE LINEA**
C*****************************************************************
C
C        PURPOSE
C
C           TO GENERATE A LEAST SQUARES POLYNOMIAL TO FIT
C           GIVEN
C           TABULAR DATA.
C
C
C        USAGE
C
C           CALL LINEA (N,X,Y,M1,W,YN)
C
C
C        DESCRIPTION OF PARAMETERS
C
C           N    - NUMBER OF TABULATED POINTS SUPPLIED (MAX.
C           300).
C           X    - VECTOR OF INDEPENDENT VALUES
C           Y    - VECTOR OF SUPPLIED DEPENDENT VALUES.
C           M1   - POWER OF THE LARGEST POLYNOMIAL TO BE
C           TRIED (MAX. 10).
C           W    - VECTOR OF WEIGHTING VALUES (SET TO UNITY
C           IF NOT USED).
C           YN   - VECTOR OF CALCULATED DEPENDENT VALUES.
C
C        REMARKS
C           VECTORS X,Y, AND W SHOULD BE DIMENSIONED IN
C           DOUBLE PRECISION
C           AND BE OF LENGTH 300.
C
C
C        SUBROUTINES REQUIRED
C
C           IBM 360 SSP SUBROUTINE DGELG  FOR THE SOLUTION
C           OF A SET OF
C           LINEAR EQUATIONS.
C
C
C        METHOD
C
C           STANDARD LEAST SQUARES TECHNIQUE  -  LAPIDUS
C           'DIGITAL
C           COMPUTATION FOR CHEMICAL ENGINEERS'  MCGRAW-HILL
C            1962
C           CHAPTER 7.
```

SUBROUTINE 'LINEA'

```
C
C
C         AUTHOR OF PROGRAM
C
C            WRITTEN BY V. KRISHNA,1967,U OF A.
C
C
C*******************************************************************
C
C

      SUBROUTINE LINEA (N,X,Y,M1,W,YN)
      DOUBLE PRECISION X(300),Y(300),R(10),A(10,10),B(300)
     1,S(10),YO(300
     1,CON(10,10),W(300),YN(300)
      DIMENSION X1(101),Y1(101),Y2(101),Y3(101)


      WRITE(6,4)
    4 FORMAT(1H1,30X,'LEAST  SQUARES  CURVE  FIT  PROGRAM'/
     1/10X,'DATA
     1UPPLIED'//23X,'X',19X,'Y',19X,'W'//)
      WRITE(6,3) (X(I),Y(I),W(I),I=1,N)
    3 FORMAT(10X,3D20.6)
      M=M1+1
      VARN=1. E 10

C     FORM NORMAL EQUATIONS
C

      DO 10 J=1,M
      R(J)=0.0
   10 CONTINUE
      DO 9 J=1,N
      R(1)=R(1)+Y(J)*W(J)
    9 CONTINUE
      DO 11 J=2,M
      DO 11 I=1,N
      IF(X(I) .LE. 0.0 ) GO TO 11
      R(J)=R(J)+Y(I)*X(I)**(J-1)*W(I)
   11 CONTINUE
      DO 12 I=1,M
      DO 12 J=1,M
      A(I,J)=0.0
   12 CONTINUE
      DO 16 J=1,M
      DO 16 K=1,M
      DO 16 I=1,N
      IF( DABS(X(I)) .LE. 0.0 ) GO TO 16
      A(K,J)=A(K,J)+X(I)**(J+K-2)*W(I)
```

SUBROUTINE 'LINEA'

```
   16 CONTINUE
      A(1,1)=0.0
      DO 17 I=1,N
      A(1,1)=A(1,1)+X(I)
   17 CONTINUE
      DO 51 K=2,M
      DO 19 J=1,K
      S(J)=R(J)
   19 CONTINUE
      DO 20 J=1,K
      DO 20 I=1,K
      IJ=(J-1)*K+I
      B(IJ)=A(I,J)
   20 CONTINUE
      KK=K*K

C     SOLVE FOR NORMAL EQUATIONS
C

      CALL DGELG(S,B,K,1,0.0001,IER)
      DO 30 J=1,N
      YO(J)=0.0
   30 CONTINUE
      DO 31 I=1,N
      IF(DABS(X(I)).GT. 0.0 ) GO TO 29
      YO(I)=S(1)
      GO TO 31
   29 CONTINUE
      DO 31 J=1,K
      YO(I)=YO(I)+S(J)*X(I)**(J-1)
   31 CONTINUE

C     COMPUTE VARIANCE
C

      VAR=0.0
      DO 32 I=1,N
      VAR=VAR+(YO(I)-Y(I))**2
   32 CONTINUE
      VAR=VAR/(N-1)
      KZ=K-1
      WRITE(6,70)KZ,VAR
   70 FORMAT(5X,'FITTING  BY  A  POLYNOMIAL  OF  ORDER',I5,'
     1 ,'//10X,
     1'WITH  A  VARIANCE  OF  ',E15.5,' ,'//10X,'AND
     1 CONSTANTS,'//)
      WRITE(6,71)(S(L),L=1,K)
   71 FORMAT(15X,D20.12/)
      IF( VAR .GE.VARN ) GO TO 40
      VARN=VAR
```

SUBROUTINE 'LINEA'

```
      KN=K
      NK=K-1
      DO 34 I=1,N
      YN(I)=YO(I)
   34 CONTINUE
      DO 33 J=1,K
      CON(K-1,J)=S(J)
   33 CONTINUE
   40 CONTINUE
   51 CONTINUE
      WRITE(6,5) NK
    5 FORMAT(1H0,'   THE  BEST  FIT  POLYNOMIAL  IS  OF
     1 DEGREE  ',I4//
      WRITE(6,6)
    6 FORMAT(10X,'OBSERVED',10X,'CALCULATED'//)
      WRITE(6,60) (Y(I),YN(I),I=1,N)
   60 FORMAT(8X,D12.5,10X,D12.5)
      RETURN
      END
```

APPENDIX IV

EQUIPMENT DETAILS

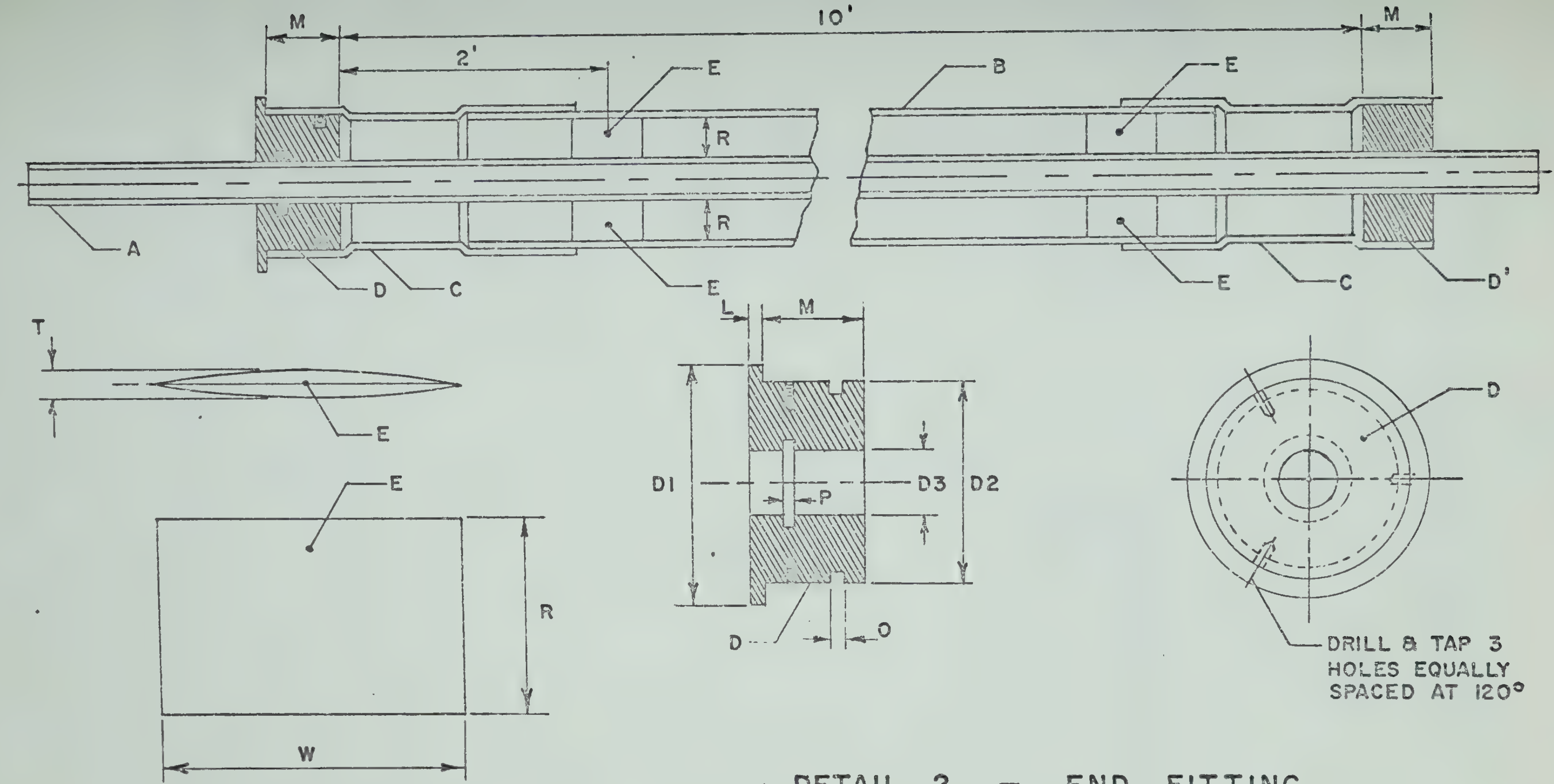
This appendix contains detailed drawings of the test heat exchanger and an address list for equipment suppliers.

The test heat exchanger drawings were used to construct three sizes of heat exchanger; however, shortness of time only permitted testing of one size, the largest.

The symbols used in Table IV B are defined in Table 1.

Figure IV A consists of a cross-sectional drawing of the test heat exchanger, and two detailed drawings of integral components. Specific dimensions which were different for all three sizes of heat exchanger, are labelled with letters. The values for these dimensions are given in Table IV A. Details of items marked with letters not referred to in Table IV A are discussed in the following material.

The heat exchanger was fabricated from two standard twelve foot lengths of Anaconda copper tubing. Tube (A) was 3/4 inch, Type H, and the shell (B) was one inch, Type H, 1¼ inch, Type DWV, or 1½ inch, Type DWV, depending upon the heat exchanger size. The inside and outside diameters of the tube were 0.815 and 0.875 inches, respectively. For the three different shell sizes, the respective inside and outside diameters were 1.061, 1.125 and 1.295, and 1.375, 1.541 and 1.625 inches. The tubing used was thin walled, and was chosen in an effort to minimize the heat capacity of the heat exchanger walls. This was done to minimize the



DETAIL 1
SPACING LUGS

DETAIL 2 - END FITTING

(NOT TO SCALE)

FIGURE IVA - DETAILS OF HEAT EXCHANGER

TABLE IV A

DIMENSIONS FOR FIGURE IV A FOR EACH OF THE THREE HEAT EXCHANGER SIZES

(All dimensions are given in inches.)

DIMENSIONS ON FIGURE IV A	HEAT EXCHANGER SIZE 3/4 x 1	3/4 x 1-1/4	3/4 x 1-1/2
R	+0.0 0.0930-0.005	+0.0 0.210-0.005	+0.0 0.326-0.005
W	3/8	3/8	3/8
T	1/16	1/16	1/16
D1	1-1/4	1-1/2	1-3/4
D2	See Figure IV-B	+0.0 1.375-0.005	+0.0 1.625-0.005
D3	See Figure IV-B	+0.005 0.875-0.0	+0.005 0.875-0.0
L	1/16	1/16	1/16
M	Tee Sleeve Length	Tee Sleeve Length	Tee Sleeve Length
O	See Figure IV-B	0.125	0.125
O (Depth)	See Figure IV-B	0.115	0.115
O-Ring Size for Shell Side	3/4 x 1 x 1/8	1-3/8 x 1-1/8 x 1/8	1-5/8 x 1-3/8 x 1/8
P	See Figure IV-B	0.125	0.125
P (Depth)	See Figure IV-B	0.115	0.115
O-Ring Size for Tube Side	3/4 x 1 x 1/8	3/4 x 1 x 1/8	3/4 x 1 x 1/8

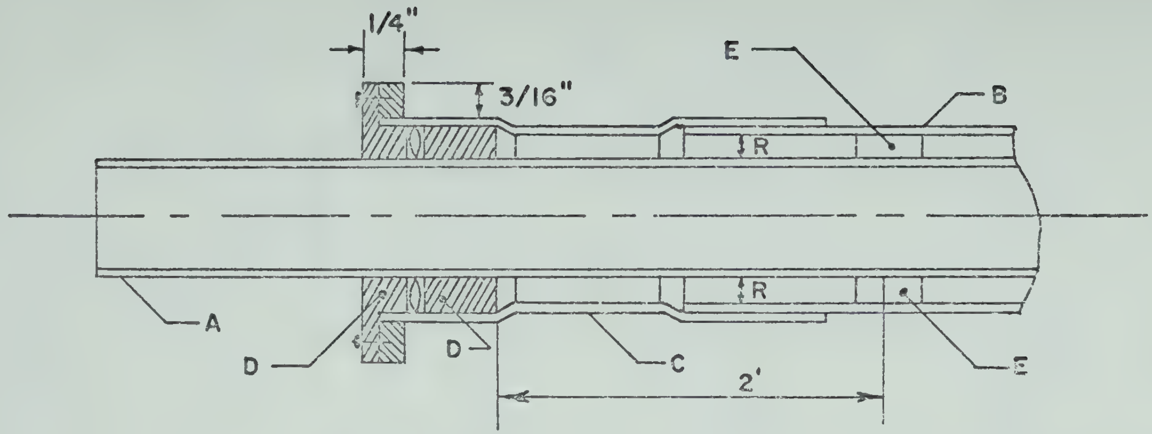
damping effect the wall capacities have on the temperature fluctuations in the fluid streams. The end fittings (C) were standard threadless copper tees with a run diameter compatible with the shell tubing, and a branch diameter of 1 inch. The tees were positioned so the shell inlet and outlet were 180 degrees apart with respect to the heat exchanger axis. The end bushings (D and D′) were machined from brass to conform with the tube OD and the tee sleeve ID. One bushing (D′) was a simple annular section which was soldered into position and the other (D) was fitted with an O-ring seal to allow for thermal expansion (See Detail II of Figure IV-A). Three screws were placed in the latter bushing to hold it in position under pressure. Because of the small annular region for the 1 inch shell size, a modification of the O-ring configuration was necessary as shown in Detail I of Figure IV-B.

To give the tube concentricity with respect to the shell, three spacing lugs (E) were soldered to the outside of tube at 120 degree angles every two feet along the axis of the heat exchanger. The shape of these lugs is shown in Detail I of Figure IV-A.

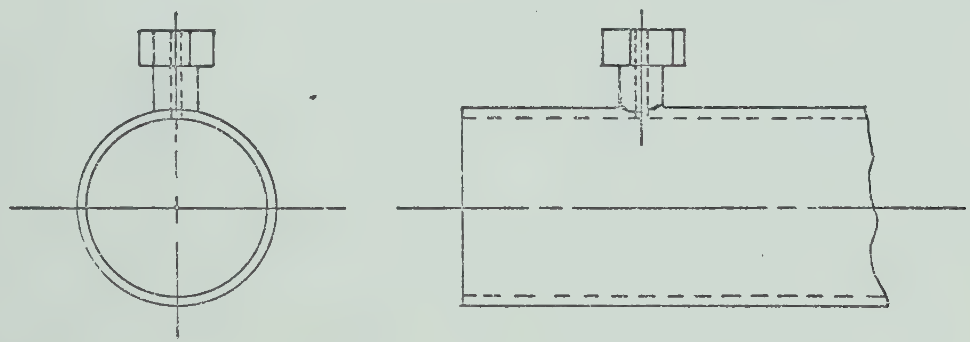
The overall length of all three test heat exchangers was ten feet. This was defined as the distance between the inner face of the bushings at each end.

Standard copper unions were fitted on to all four branches of the heat exchanger as close to the exchanger as possible. This facilitated easy changes from one heat exchanger size to the next.

Thermocouple probes were mounted at convenient points near the inlet and outlet of each stream. They were the ceramic packed type with a 1/16 inch OD sheath (see Table 1) which is suitable for mounting in a



DETAIL 1
END FITTING FOR 3/4 x 1 HEAT EXCHANGER



DETAIL 2
THERMOCOUPLE MOUNTING

(NOT TO SCALE)

FIGURE IV B - DETAILS OF HEAT EXCHANGER

compression type fitting. Detail II of Figure IV-B shows how these fittings were mounted on the exchanger. The procedure for mounting was as follows:

a. The bottom face of the fitting was filed to a concave curvature that approximately matched the convex curvature of the exchanger wall.

b. The fitting (brass) was then soldered on to the surface of the exchanger at the desired location.

c. A 1/16 drill was used to drill a hole in the exchanger wall that matched the location of the fitting. With this type of installation, the thermocouple could easily be inserted into the stream to any depth and held in position by tightening the compression nut. To assure that the depth of the thermocouple would be continuously adjustable, teflon ferrules were used in the compression fittings to prevent damaging the sheath surfaces.

For dynamic temperature measurement, two thermopiles were used; one in each outlet stream. They were positioned in one end of a copper tee with flow entering the tee along the run at the opposite end and leaving the tee via the branch. To achieve accurate measurement of the stream temperatures, the thermopile tee was positioned as near the exchanger body as possible. The limiting factor in this regard was the union referred to earlier which had to come between the heat exchanger and the thermopile tee so that changing of heat exchangers would not disturb the thermopiles. This short distance did not yield a temperature drop because experimental work revealed that heat transport was such that no measurable temperature difference could be detected less than five feet down stream of the heat exchanger outlets.

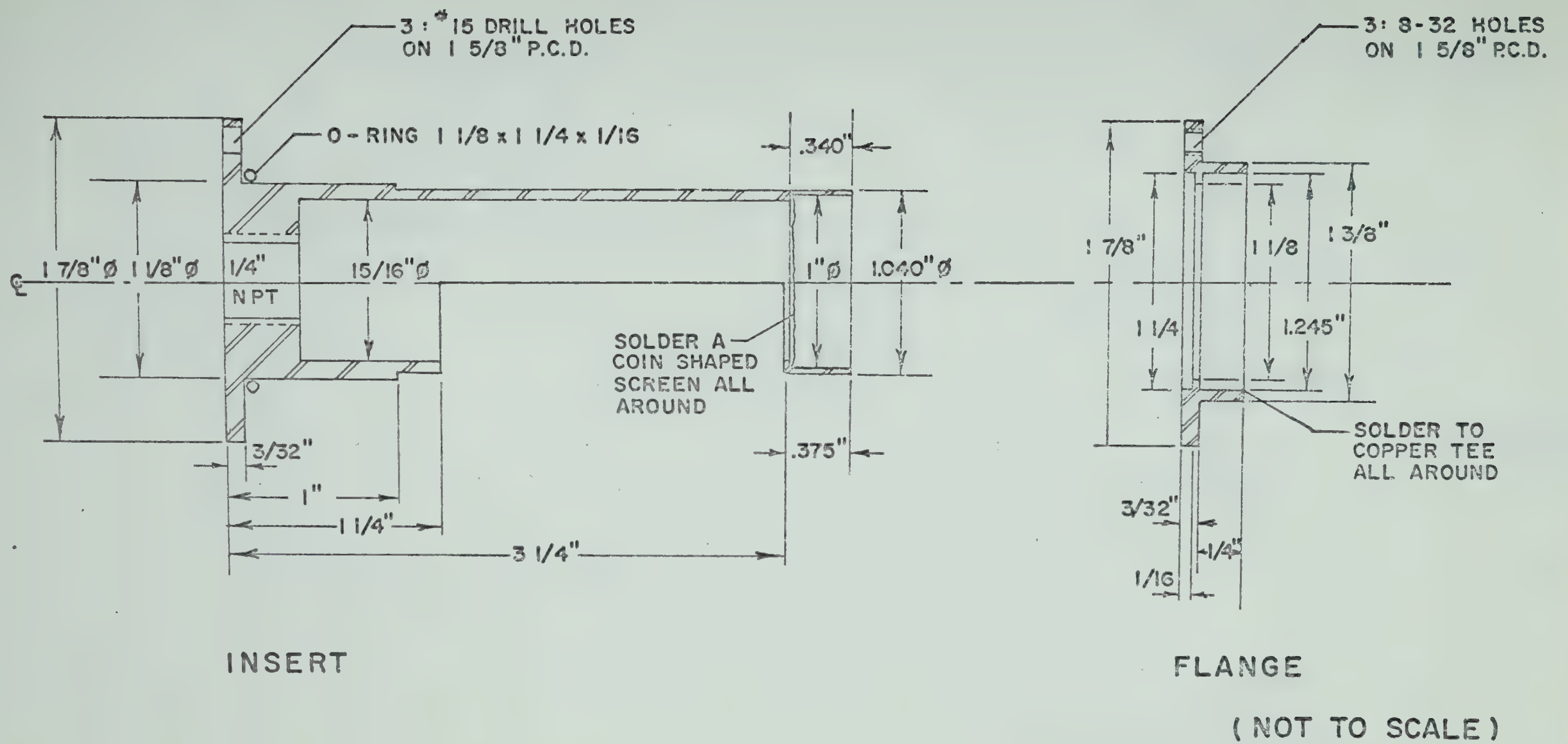


FIGURE IV C - THERMOPILE SUPPORT (SHELL)

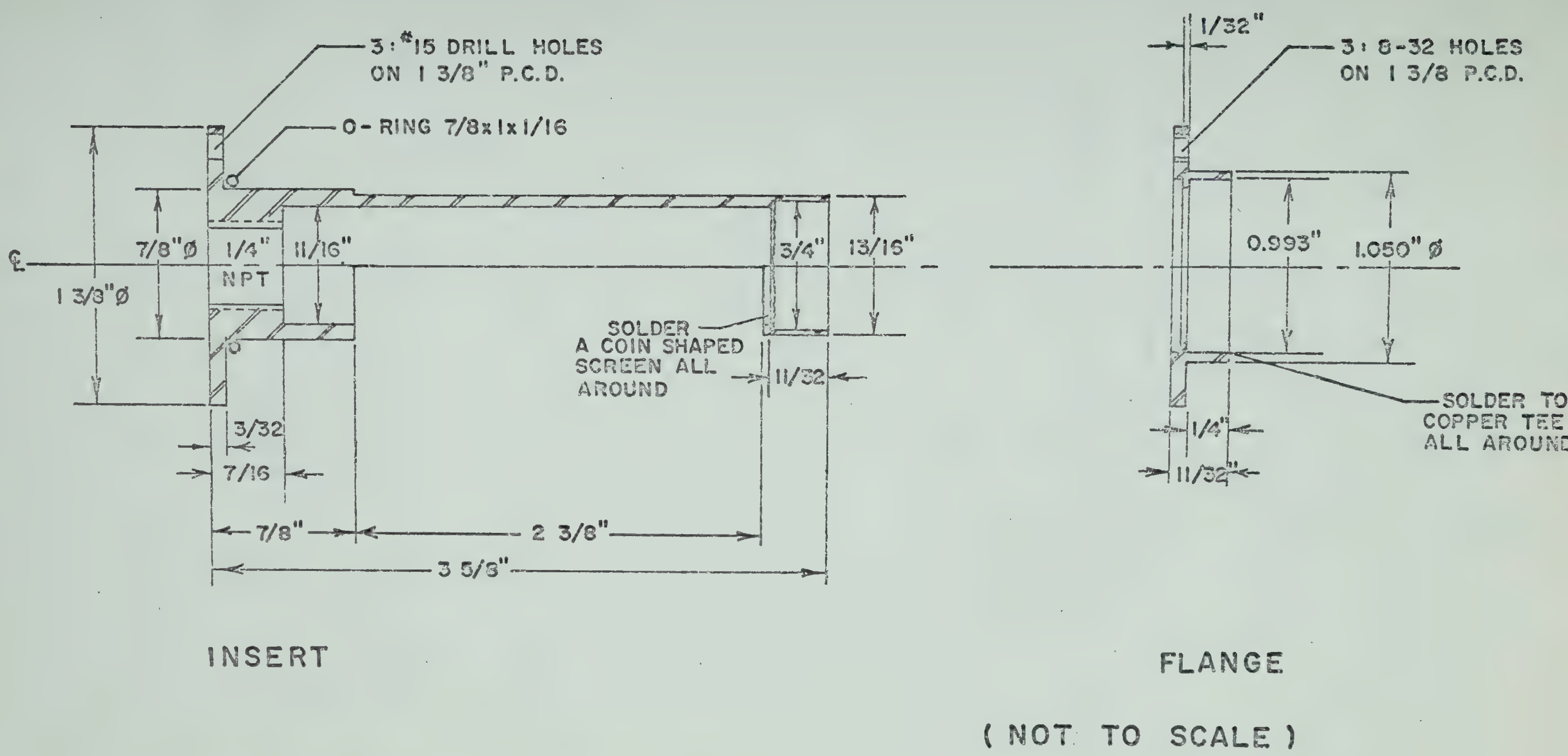


FIGURE IV D - THERMOPILE SUPPORT (TUBE)

The thermopiles used were supplied with the probe wires mounted inside a ¼ inch diameter, 4 inch long stainless steel sheath designed to be positioned in a ¼ inch male NPT compression fitting. Figures IV-C and IV-D show the supports that were fabricated for mounting the thermopiles in the tees. Two sizes of support were made, 1 inch for the shell and 3/4 inch for the tube.

As mentioned in Section 5.2.2, these supports perform a secondary function by supporting a loosely packed piece of gauze upstream of the thermopile which served to mix the fluid and minimize thermal noise. To prevent the gauze from being carried downstream with the flow, a coin shaped piece of brass screen was soldered to the support at the location shown in Figures IV C and IV D.

The support fits to a closure tolerance inside the tee, and is held in position by three screws which thread into the flange shown in Figure IV C and IV D. The flange is soldered to the outer surface of the copper tee. The compression fitting is threaded into the ¼ inch female NPT and the thermopile is positioned for the desired probe contact with the stream fluid, then the compression nut is tightened to secure the probe.

As mentioned in Section 5.2.2, the thermopiles can be interchanged from one stream to the other. This is achieved by loosening the compression, removing one probe, replacing it with the other and re-tightening.

TABLE IV B

LIST OF EQUIPMENT SUPPLIERS

SYMBOL	MANUFACTURER
BV1 BV2	Whitey Research Tool Co. 5525 Marshall St. Oakland 8, California
C1 C2	University of Alberta Technical Services Department Electronic Division University of Alberta Edmonton, Alberta
CR1 CR2	Canadian General Electric Co. Ltd. 14325 - 112 Avenue Edmonton, Alberta
FC1 FC2 FC5	Spartan Controls Ltd. 5638 - 103A Street Edmonton, Alberta
FCSA	Honeywell Controls Ltd. 11820 - 142 Street Edmonton, Alberta
FT1 FT2	Flow Technology, Inc. 401 South Hayden Road Tempe, Arizona 85281

TABLE IV B (CONT'D)

LIST OF EQUIPMENT SUPPLIERS

SYMBOL	MANUFACTURER
FV1 FV2	University of Alberta Technical Services Department Electronics Division University of Alberta Edmonton, Alberta
M1 M2	Control Company Canada, Ltd. Mfrs. of Redmond Electric Motors St. Thomas, Ontario
OV1 OV2	Western Alloys and Metals Ltd. 1475 Boundary Road Vancouver 6, B.C.
P1 P2	Robert Morse Corporation Ltd. 10940 - 120 Street Edmonton, Alberta
R1 R2	Canadian Electronics Ltd. 109 Street and 107 Avenue Edmonton, Alberta Manufactured by: Superior Electric Co. Bristol, Connecticut

TABLE IV B (CONT'D)

LIST OF EQUIPMENT SUPPLIERS

SYMBOL	MANUFACTURER
S1	Crane Supply
S2	11925 Kingsway Avenue
S3	Edmonton, Alberta
S4	Manufactured by: Keckley Ltd. 3400 Cleveland Street Skokie, Illinois
SB1	Canadian General Electric Co. Ltd.
SB2	14325 - 112 Avenue
	Edmonton, Alberta
SH1	Flow Technology, Inc.
SH2	401 South Hayden Road,
	Tempe, Arizona 85281
SHX	J. R. Stephenson Ltd. 14600 - 116th Avenue Edmonton, Alberta Manufacturer: Taco Heaters Canada Ltd. Toronto, Ontario

TABLE IV B (CONT'D)

LIST OF EQUIPMENT SUPPLIERS

SYMBOL	MANUFACTURERS
STA	J. R. Stephenson Ltd.
STB	14606 - 116th Avenue
	Edmonton, Alberta
	Manufacturer: Sarco Canada Ltd. Markham Road and MacDonald- Cartier Freeway RR 1 Agincourt, Ontario
T1	Thermo-Electric (Canada) Ltd.
T2	P. O. Box 10
	Brampton, Ontario
TP1	Science Products Corporation
TP2	Route 46
	Dover, N.J. 07801